СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛ. ОХРИДСКИ“

СТОПАНСКИ ФАКУЛТЕТ

ИГРОВИ МОДЕЛИ И МОДЕЛИРАНЕ НА ВРЕМЕВИ РЕДОВЕ

АВТОРЕФЕРАТ

За присъждане на образователна и научна степен „доктор”

Област на висше образование: 3. Социални, стопански и правни науки

Професионално направление: 3.8. Икономика

Научно направление: Аналитични изследвания върху данни /Data Science/

Разработил:

Владислав Красимиров Танов

Научен ръководител :

доц. д-р Николай Нетов

София

2023

СЪДЪРЖАНИЕ

Увод

Теорията на игрите е концепция, която учените могат да овладеят, за да предскажат как рационалните хора ще вземат решения, които им помагат да направят ефективни решения, основаващи се на данни, при стратегически обстоятелства.

Учените в областта на данните могат да прилагат теорията на игрите въз основа на типа на проблема за вземане на решения, с който се занимават: https://www.dezyre.com/article/is-game-theory-important-for-data-scientists/139

Игровото моделиране и моделирането на големи данни намира общи точки на съвместни приложения в гео-науките и гео-данните. Авторът (Bruce, 2013) очертава механизмите за прилагане на моделите на теорията на игрите за големи анализи на данни и вземане на решения в областта на гео-науките и гео-данните. Авторът предлага използването на стратегически, конкурентни модели на теорията на игрите за целите на групирането на спектрални ленти при използване на хиперспектрални изображения. Предложената система използва филтриране на конфликтни данни, базирано на взаимна ентропия и процес на взаимодействие на стратегии на множество групи в конфликтна среда, чиято цел е да се увеличи максимално ползата от множество групи от цялата система. Предложената система използва равновесието на Nash като средство за намиране на стабилизиращо решение на проблема с групирането на групите и реализира модела при предположението, че всички играчи са рационални. Авторът използва предложеното групиране на групи като компонент в система за мултифункционално решение за сливане (MCDF) за автоматично класифициране на земно покритие с хиперспектрални изображения.

От тази гледна точка ние ще разгледаме някои различни видове игри и представяме методологии за търсене на равновесие чрез решаване на Рикатиеви уравнения. Чрез разгледаните видове игри показваме как в различни условия може да се намери равновесна партия в конкурентна среда. Използваме стабилизиращи решения за уравненията на Рикати за достигане до равновесието на Наш: https://www.igi-global.com/chapter/applied-game-theory-in-business-analytics/107224

Първа и втора глава от настоящия проект на дисертационен труд са посветени на намиране на равновесие в линейноквадратични игри чрез създаване на методи и алгоритми за търсене на стабилизиращи решения на съответни Рикатиеви уравнения. Тези изследвания биха могли да бъдат в основата на разработване на игрови модели с приложения при анализ на големи данни (machine learning.) Изследвания в тази посока се появяват при прилагане на концепцията за намиране на оптималните стратегии на Неш в условията на класификационна задача (Мои съавтори, 2023; Wang и съавтори, 2019).

В първа глава разглеждаме линейноквадратична стохастична игра, анализирана от Zhu и Zhang, за която построяваме итерационен метод за намиране на стабилизиращо решение на система от четири нелинейни матрични уравнения. В първа глава са използвани методии, алгоритми и примери от (Иван Иванов, 2012; Ивелин Иванов, 2016). В същото време предложените методи допълват изследванията в (Ивелин Иванов, 2016).

Във втора глава разглеждаме антаготистични игри и игрови модели върху позитивни системи. Предложени са методи за намиране на стабилизиращо неотрицателно решение на съответно Рикатиево уравнение. Резултатите са публикувани в три статии, две от които са индексирани в Скопус.

Авторите (Koziarski и други,2020) и много други анализират трудностите при моделиране на многокласови и небалансиране големи данни. Целта в трета глава е да развием ориентиран към данните подход за провеждане на класификационен анализ на големи данни. Формулираме оптимизационен модел, който търси най-добро тренировъчно множество, в конкретен смисъл, за модели, провеждащи класификационен анализ. За решаване на оптимизационната задача предлагаме алгоритъм, който е приложен към различни множества от големи данни. Резултатите са докладвани на международна конференция и са подкрепени с две публикации, индексирани в Скопус (https://www.scopus.com/authid/detail.uri?authorId=57208207140).

Означенията в авторефератa запазват номерацията и цитиранията, съгласно текста на дисертационния труд.

Глава първа. Линейноквадратични диференциални стохастични игри

Линейните квадратични игри, в които решаването на Рикатиеви уравнения с цел достигане на равновесни партии са широко изследвани в научната литература. Пример за такива изследвания може да цитираме (Azevedo-Perdicoulis, Jank, 2005), (Basar, Olsder, 1999), (Broek, Engwerda, Schumacher, 2003), (Engwerda 2005). Основна роля в теорията се отделя на стохастичните диференциални игри, в които се използва лемата на Ито, прилагана за диференциални системи със смущения в състоянието на системата и управлението (Yu, 2012), (Zhu, Zhang, 2013), (Zhu, Zhang, Bin, 2014). Интензивното изучаване на обобщените Рикатиеви уравнения се базира на тяхната широка приложимост – например при изследване и построяване на оптимални финансови портфейли се преминава през решаването на подходящо обобщено Рикатиево уравнение – (Yao, Zhang, Zhou, 2006; Costa, de Paulo 2007; Costa, de Oliveira, 2012).

В тази глава представяме концепцията за теорията на игрите и нейната употреба като инструмент за вземане на решения в конкурентна ситуация сред играчите. Търсим равновесната точка чрез търсене на стабилизиращо решение за система от две Рикатиеви уравнения и още две допълнителни уравнения, както са ги извели авторите в (Zhu, Zhang, 2013).

Целта в настоящата глава е да опишем и предложим методи и алгоритми за търсене на стабилизиращо решение на система от Рикатиеви уравнения, които решения водят по намиране на равновесната точка в разглеждания игрови модел.

1.2. Линейноквадратични диференциални стохастични игри със смущения

Разглеждаме линейноквадратична стохастична игра, изследвана от Zhu и Zhang в тяхната статия в (Zhu, Zhang, 2013). Съществуването на равновесие в терминологията на решение на система от матрични уравнения е представено от авторите в Теорема 2 (Zhu, Zhang, 2013). Авторите Zhu и Zhang не представят метод или алгоритъм за намиране на решение на тези уравнения. Ние ще предложим подход за пресмятане на търсеното решение. Разглежданата система Рикатиеви уравнения, всъщност търси решние, което задава равновеси ена Наш в стохастична диференциална игра, описана от Zhu и Zhang. Предложеният тук подход позволява да се разглежда стохастична диференциална игра с различен брой играчи. Предложеният метод не зависи от броя на играчите.

В настоящата секция ще опишем алгоритъм публикуван (Ivelin Ivanov и V. Tanov, 2018, (Ivelin Ivanov и V.Tanov, 2018, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise) за намиране на съответното решение. Съгласно Zhu и Zhang, равновесието по Наш е решението $\widetilde{X_1}, \widetilde{X_2}$ на следната система от две свързани нелинейни матрични уравнения:

$$R_1(\mathrm{X}_1, \mathrm{X}_2) := \mathrm{X}_1\bar{A}_0 + \bar{A}_0{}^T\mathrm{X}_1 + \bar{A}_1{}^T\mathrm{X}_1\bar{A}_1 + \bar{Q}_1 - \left(\mathrm{X}_1\mathrm{B}_1 + \bar{A}_1{}^T\mathrm{X}_1\mathrm{C}_1\right)$$

$$\times (R_{11} + C_1^T X_1 C_1)^{\{-1\}}(B_1^T X_1 + C_1^T X_1 \bar{A}_1) \ = 0$$

$$F_1 = -(R_{11} + C_1^T X_1 C_1)^{\{-1\}}(B_1^T X_1 + C_1^T X_1 \bar{A}_1)$$

$$(R_{11} + C_1^T X_1 C_1) > 0 \qquad (1.4)$$

$$R_2(\mathrm{X}_1, \mathrm{X}_2) = \mathrm{X}_2 \tilde{A}_0 + \tilde{A}_0{}^T \mathrm{X}_2 + \tilde{A}_1{}^T \mathrm{X}_2 \tilde{A}_1 + \bar{Q}_2$$

$$-\left(\mathrm{X}_2 \mathrm{B}_2 + \tilde{A}_1{}^T \mathrm{X}_2 \mathrm{C}_2\right)(R_{22} + C_2^T X_2 C_2)^{\{-1\}}\left(B_2^T X_2 + C_2^T X_2 \tilde{A}_1\right) = 0$$

$$F_2 = -(R_{22} + C_2^T X_2 C_2)^{\{-1\}}\left(B_2^T X_2 + C_2^T X_2 \tilde{A}_1\right)$$

$$(R_{22} + C_2^T X_2 C_2) > 0 \quad \text{label\{H17Eq.10\}} \quad (2.4)$$

при следните означения

$$\bar{A}_0 = \mathrm{A}_0 + \mathrm{B}_2 \mathrm{F}_2, \quad \bar{A}_1 = \mathrm{A}_1 + \mathrm{C}_2 \mathrm{F}_2,$$

$$\tilde{A}_0 = \mathrm{A}_0 + \mathrm{B}_1 \mathrm{F}_1, \quad \tilde{A}_1 = \mathrm{A}_1 + \mathrm{C}_1 \mathrm{F}_1,$$

$$\bar{Q}_1 = \mathrm{Q}_1 + \mathrm{F}_2^{\mathrm{T}} R_{12} \mathrm{F}_2, \quad \bar{Q}_2 = \mathrm{Q}_2 + \mathrm{F}_1^{\mathrm{T}} R_{21} \mathrm{F}_1.$$

Освен това A_0 и A_1 са реални nxn матрици, Q_1и Q_2 са реални симетрични nxn матрици, B_1 и C_1 са реални nxm_1 матрици , B_2 и C_2 са реални nxm_2 матрици, R_{11} и R_{21} са реални m_1xm_1 матрици, и R_{12} и R_{22} са реални m_2xm_2 матрици.

Известни са следните определения. Една матрица A се нарича устойчива, ако всички нейни собствени стойности са разположени в лявата полуравнина, относно ординатната ос. Ще използваме записа X > Y или $X \geq Y$, ако X – Y е положително определена матрица, т.е. с положителни собствени стойности или X – Y е положително полуопределена матрица, т.е. с неотрицателни собствени стойности.

Представяме експериментални резултати, които показват, че предложеният теоретико-алгоритмичен подход на играта значително води до търсеното равновесие.

1.3. Итерационен метод

Построяваме итерационен алгоритъм за решаване на дефинираната система от нелинейни матрични уравнения и неравенства (1.4). Построява се матрична редица от решения на тази система, която е сходяща към стабилизиращо решение за две Рикатиеви уравнения. Това стабилизиращо решение води намиране на равновесие по Наш за разглежданата игра. Представяме експериментални резултати, които показват, че предложеният теоретико-алгоритмичен подход на играта води до търсеното равновесие.

Записваме Рикатиевите уравнения $R_1(\mathrm{X}_1, \mathrm{X}_2) = 0$ и $R_2(\mathrm{X}_1, \mathrm{X}_2) = 0$ като общо Рикатиево уравнение но с по-голяма размерност

$$\mathcal{R}(\boldsymbol{X}) = \mathcal{A}_0{}^T \mathbf{X} + \mathbf{X}\, \mathcal{A}_0 + \Pi_1(\mathbf{X}) + Q$$

$$-\mathcal{S}(\mathbf{X})\, [\Re(\mathbf{X})]^{\{-1\}}[\mathcal{S}(\mathbf{X})]^{\mathrm{T}} = \mathbf{0} \qquad (1.5)$$

където

$$\Re(\mathbf{X}) = R + \mathcal{C}^T \mathbf{X}\, \mathcal{C} = diag(R_{11} + C_1^T X_1 C_1, R_{22} + C_2^T X_2 C_2)$$

$$\mathcal{S}(\mathbf{X}) = \mathbf{X}\, \mathcal{B} + \mathcal{A}_1{}^T \mathbf{X}\, \mathcal{C}$$

$$= \mathrm{diag}\,(\mathrm{X}_1 \mathrm{B}_1 + \bar{A}_1^{\mathrm{T}} \mathrm{X}_1 \mathrm{C}_1, \mathrm{X}_2 \mathrm{B}_2 + \tilde{A}_1^{\mathrm{T}} \mathrm{X}_2 \mathrm{C}_2)$$

$$\Pi_1(\mathbf{X}) = \mathcal{A}_1{}^T \mathbf{X}\, \mathcal{A}_1 = \mathrm{diag}\,(\bar{A}_1^{\mathrm{T}} \mathrm{X}_1 \bar{A}_1, \tilde{A}_1^{\mathrm{T}} \mathrm{X}_2 \tilde{A}_1)$$

$$\mathcal{A}_0 = \mathrm{diag}\,(\bar{\mathrm{A}}_0, \tilde{\mathrm{A}}_0); \qquad \mathcal{A}_1 = \mathrm{diag}\,(\bar{\mathrm{A}}_1, \tilde{\mathrm{A}}_1);$$

$\mathcal{B} = \text{diag}(\mathrm{B}_1, B_2)$; $\mathcal{C} = \text{diag}(\mathrm{C}_1, C_2)$;

$R = \text{diag}(\mathrm{R}_{11}, \mathrm{R}_{22})$; $\mathcal{Q} = \text{diag}(\bar{Q}_1, \bar{Q}_2)$;

$\mathbf{X} = \text{diag}(\mathrm{X}_1, X_2)$;

Разглежданото уравнение на Рикати (1.5) е от същия вид, както уравнението (1.1). Реализираната тук идея е да използваме итерационния метод на Ляпунов (1.3) към системата уравнения (1.5). За начална матрица избираме $\mathrm{X}^{(0)} = \left[\mathrm{X}_1^{(0)}, \mathrm{X}_2^{(0)}\right]$ и пресмятаме

$$
\begin{aligned}
F_1^{(0)} &= -\left(R_{11} + C_1^T X_1^{(0)} C_1\right)^{-1}\left(B_1^T X_1^{(0)} + C_1^T X_1^{(0)} \bar{A}_1\right) \\
\tilde{A}_1 &= \mathrm{A}_1 + \mathrm{C}_1 F_1^{(0)} \\
F_2^{(0)} &= -\left(R_{22} + C_2^T \mathrm{X}_2^{(0)} C_2\right)^{-1}\left(B_2^T \mathrm{X}_2^{(0)} + C_2^T \mathrm{X}_2^{(0)} \tilde{A}_1\right) \qquad (1.6) \\
\bar{A}_1 &= \mathrm{A}_1 + \mathrm{C}_2 F_2^{(0)} \\
F_1^{(0)} &= -\left(R_{11} + C_1^T X_1^{(0)} C_1\right)^{-1}\left(B_1^T X_1^{(0)} + C_1^T X_1^{(0)} \bar{A}_1\right)
\end{aligned}
$$

Построяваме следната матрична редица $\{\mathbf{X}^{(\mathbf{k})}\}_{\mathbf{0}}^{\infty}$ по следния начин. Предполагаме, че е известна матрицата $\mathbf{X}^{(\mathbf{k})}$. Ще пресметнем $\mathbf{X}^{(\mathbf{k+1})}$ като последователно пресмятаме

$$
\begin{aligned}
&\tilde{A}_1 = \mathrm{A}_1 + \mathrm{C}_1 F_1^{(k-1)}; \qquad \bar{A}_1 = \mathrm{A}_1 + \mathrm{C}_2 F_2^{(k-1)} \\
&\mathcal{A}_1 = \text{diag}(\bar{\mathrm{A}}_1, \tilde{\mathrm{A}}_1) \\
&\mathcal{S}\left(\mathbf{X}^{(\mathbf{k})}\right) = \mathbf{X}^{(\mathbf{k})}\mathcal{B} + \mathcal{A}_1{}^T \mathbf{X}^{(\mathbf{k})}\mathcal{C} \\
&\mathcal{F}_{\mathbf{X}^{(\mathbf{k})}} = \left[\Re\left(\mathbf{X}^{(\mathbf{k})}\right)\right]^{-1}\left[\mathcal{S}\left(\mathbf{X}^{(\mathbf{k})}\right)\right]^{\mathrm{T}} = \\
&\qquad = \text{diag}\left(F_1^{(k)}, F_2^{(k)}\right) = \\
&\qquad = \text{diag}\left(F_1\left(\mathbf{X}^{(\mathbf{k})}\right), F_2\left(\mathbf{X}^{(\mathbf{k})}\right)\right) \\
&\tilde{A}_0 = \mathrm{A}_0 + \mathrm{B}_1 F_1^{(k)} \quad \bar{A}_0 = \mathrm{A}_0 + \mathrm{C}_2 F_2^{(k)}, \qquad (1.7) \\
&\mathcal{A}_0 = \text{diag}(\bar{\mathrm{A}}_0, \tilde{\mathrm{A}}_0) \\
&\bar{Q}_1 = \mathrm{Q}_1 + \left(F_2^{(k)}\right)^T R_{12} F_2^{(k)} \; ; \quad \bar{Q}_2 = \mathrm{Q}_2 + \left(F_1^{(k)}\right)^T R_{21} F_1^{(k)} \\
&\mathcal{Q} = \text{diag}(\bar{\mathrm{Q}}_1, \bar{\mathrm{Q}}_2)
\end{aligned}
$$

След тези означения прилагаме следващия итерационен метод

$$
\begin{aligned}
&M(\mathbf{X}^{(\mathbf{k})}) = \left(\mathcal{A}_0 + \mathcal{B}\mathcal{F}_{\mathbf{X}^{(\mathbf{k})}}\right)^{\mathrm{T}} \mathbf{X}^{(\mathbf{k+1})} + \mathbf{X}^{(\mathbf{k+1})}\left(\mathcal{A}_0 + \mathcal{B}\,\mathcal{F}_{\mathbf{X}^{(\mathbf{k})}}\right) \\
&+ \mathbf{T}_{(\mathbf{X}^{(\mathbf{k})})} + \Pi_{(\mathbf{X}^{(\mathbf{k})})}\left(\mathbf{X}^{(\mathbf{k})}\right) = \mathbf{0} \qquad (1.8)
\end{aligned}
$$

при означенията

$$
\mathbf{T}_{(\mathbf{Z})} = \begin{pmatrix} \boldsymbol{I} \\ \mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})} \end{pmatrix}^{\boldsymbol{T}} \begin{pmatrix} \boldsymbol{Q} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} \\ \mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})} \end{pmatrix}
$$

$$
\Pi_{(\mathbf{X}^{(\mathbf{k})})}\left(\mathbf{X}^{(\mathbf{k})}\right) = \begin{pmatrix} \boldsymbol{I} \\ \mathcal{F}_{\mathbf{X}^{(\mathbf{k})}} \end{pmatrix}^{\boldsymbol{T}} \begin{pmatrix} \mathcal{Q} & \mathbf{0} \\ \mathbf{0} & R \end{pmatrix} \begin{pmatrix} \boldsymbol{I} \\ \mathcal{F}_{\mathbf{X}^{(\mathbf{k})}} \end{pmatrix}.
$$

При предложението, че $\mathcal{A}_0$, $\mathcal{A}_1$, $\mathcal{Q}$ са дадени матрици, то сходимостта на итерационната формула (2.8) и нейните свойства са изведени от следващата теорема

Теорема 1.3 (обобщение на Теорема 1.2) Нека съществуват симетрични матрици $\widehat{\mathbf{X}}$ и $\mathbf{X^{(0)}}$, за които $\mathcal{R}(\widehat{\mathbf{X}}) \geq \mathbf{0}$ и $\mathbf{X^{(0)}} > \widehat{\mathbf{X}}$, $\mathcal{R}(\mathbf{X^{(0)}}) < \mathbf{0}$ и $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\mathbf{X^{(0)}})}$ е устойчива матрица, където $\mathbf{F}_{(\mathbf{X^{(0)}})} = \left[\mathbf{R}\left(\mathbf{X^{(0)}}\right)\right]^{\{-1\}}\left[\mathbf{S}\left(\mathbf{X^{(0)}}\right)\right]^{\mathrm{T}}$. При тези условия матричната редица $\{\mathbf{X^{(k)}}\}_{\mathbf{0}}^{\infty}$, построена чрез формулата (1.8) притежава следните свойства:

(i) $\mathbf{X^{(s)}} > \mathbf{X^{(s+1)}}$, $\mathbf{X^{(s)}} > \widehat{\mathbf{X}}$ и $\mathcal{R}(\mathbf{X^{(s)}}) < 0$, s=0,1,2, ...

(ii) $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\mathbf{X^{(s)}})}$ е устойчива матрица за s=0,1,2, ...

(iii) $\lim_{\{\mathbf{s}\to\infty\}} \mathbf{X^{(s)}} = \widetilde{\mathbf{X}}$ е решението на Рикатиевото уравнение $\mathcal{R}(\mathbf{X}) = \mathbf{0}$ със свойството $\widetilde{\mathbf{X}} \geq \widehat{\mathbf{X}}$. Освен това, ако $\mathbf{X^{(0)}} \geq \mathbf{X}$ за всички решения $\mathbf{X}$ на $\mathcal{R}(\mathbf{X}) = \mathbf{0}$, тогава $\widetilde{\mathbf{X}}$ е максимално решение.

(iv) Собствените стойности на матрицата $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\widetilde{\mathbf{X}})}$ са разположени в затворената лявата полуравнина спрямо ординатната ос (включва се ординатната ос). Ако $\mathcal{R}(\widehat{\mathbf{X}}) > \mathbf{0}$, то собствените стойности $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\widetilde{\mathbf{X}})}$ са в лявата отворена полуравнина спрямо ординатната ос.

Ще проведем експерименти с няколко примера за намиране на стабилизиращото решение на Рикатиевото уравнение (1.5), като за целта използваме въведения итерационен метод чрез формулите (1.6)-(1.8). В експерименталната част използваме Anaconda средата с Python 3.7. Представяме формула (1.8) в по-удобен вид за програмиране и изпълнение на итерацията:

$$\left(\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\mathbf{X^{(k)}})}\right)^{\mathrm{T}} \mathbf{X^{(k+1)}} + \mathbf{X^{(k+1)}}\left(\mathcal{A}_0 + \mathbf{B}\,\mathrm{F}_{(\mathrm{X^{(k)}})}\right) + \mathcal{Q} \qquad (1.9)$$
$$+\left(\mathbf{F}_{(\mathbf{X^{(k)}})}\right)^{\mathrm{T}} \mathbf{R}\,\mathbf{F}_{(\mathbf{X^{(k)}})} + \left(\mathcal{A}_1 + \mathrm{C}\,\mathbf{F}_{(\mathbf{X^{(k)}})}\right)^{\mathrm{T}} \mathbf{X^{(k)}}\left(\mathcal{A}_1 + \mathrm{C}\,\mathbf{F}_{(\mathbf{X^{(k)}})}\right) = 0$$

Итерационната формула (1.9) се нарича формула на Ляпунов, тъй като на всяка стъпка се решава матрично уравнение на Ляпунов относно неизвестното $\mathbf{X^{(k+1)}}$. В дисертационния труд е представен алгоритъм за реализация на итерационна формула (1.9).

Ще представем един пример, реализрн по този алгоритъма. Матричните коефициенти на системата (1.4) са представени в терминологията на Python за всеки пример.

Пример 1.1.

```
import numpy as np
n=3
m1=2
m2=3
A0 = np.matrix([[-1.5, 0.17,-0.049],[0.07, -1.42, -0.027],[0.04, -0.11,-1.47]])
A1 = np.matrix([[0.7, 0.19,-0.04],[0.24, 0.9,0.9],[0.3, 0.1,0.15]])
Q1=0.3*np.matlib.identity(n)
Q2=0.025*np.matlib.identity(n)
```

```
B1= np.matrix([[0.0, 0.],[0.05, 0.1],[0.04, 0.15]]);
C1= np.matrix([[0., 0.1],[1.1, 0],[0., 0.02]]);
B2= np.matrix([[0.1, 0.5 , 0.4],[0., 0, 0.08],[0., 0., 2.2]])
C2 = np.matrix([[0.1, 0. , 0.],[0., 1.5, 0.0],[0.1, 0.05, 0.0]])
R11 = np.matlib.identity(m1);
R11[0,0]=4.0
R11[m1-1,m1-1]=5.0
R21 = np.matlib.identity(m1)/2.
R21[1,1]=10.
R22 = np.matlib.identity(m2)
R22[0,0]=2.
R22[m2-1,m2-1]=8.
R12 = np.matlib.identity(m2)/2.
R12[1,1]=2.
R12[m2-1,m2-1]=3.
```

При изпълнение на Пример 1.1 конкретните стойности са n=3, tol=1.0e-8. Избираме $X_1^{(0)}$ = diag [6,6,6], $X_2^{(0)}$ = diag [9,9,9]. При тези стойности проверяваме дали са изпълнени условията на Теоремата, а именно $R_1\left(X_1^{(0)}, X_2^{(0)}\right) < 0$, $R_2\left(X_1^{(0)}, X_2^{(0)}\right) < 0$. За долна граница на матричните редици избираме $\hat{X}_1 = \hat{X}_2 = \text{diag}\,[0.0002, 0.0002, 0.0002]$, и $R_1\left(\hat{X}_1, \hat{X}_2\right) > 0$, $R_2\left(\hat{X}_1, \hat{X}_2\right) > 0$. Освен това матрицата $\mathcal{A}_0, + \mathbf{B}\,\mathbf{F}_{(\mathbf{X}^{(0)})}$ е устойчива, т.е. нейните собствени стойности мат реални части в лявата полуравнина. И следователно условията на цитираната теорема са изпълнени и може да приложим итерационната формула (1.9) при така избраните матрици $\left(X_1^{(0)}, X_2^{(0)}\right)$.

За двете решения $\tilde{X}_1, \tilde{X}_2$ (които са 3x3 матрици) получаваме

$$\tilde{X}_1 = \begin{pmatrix} 0.13952043 & 0.04144027 & 0.02188102 \\ 0.04144027 & 0.15624824 & 0.03732627 \\ 0.02188102 & 0.03732627 & 0.14154421 \end{pmatrix},$$

$$\tilde{X}_2 = \begin{pmatrix} 0.0120035 & 0.003909 & 0.00222309 \\ 0.003909 & 0.01359531 & 0.0036183 \\ 0.00222309 & 0.0036183 & 0.01226303 \end{pmatrix}.$$

Решенията $\tilde{X}_1, \tilde{X}_2$ се получават след 25 итерационни стъпки на формула (1.9) и притежават свойствата, изведени в теоремата – матрицата $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\tilde{\mathbf{X}})}$ е устойчива. Но ние търсим равновесието по Наш, което се получава след като пресметнем матриците $F_1\ (\tilde{\mathbf{X}}), F_2\ (\tilde{\mathbf{X}})$:

$$F_1\ (\tilde{\mathbf{X}}) = \begin{pmatrix} -0.02010912 & -0.04090623 & -0.0385815 \\ -0.00398803 & -0.00569488 & -0.00583653 \end{pmatrix}$$

$$F_2\left(\widetilde{\mathbf{X}}\right) = \begin{pmatrix} -0.00138726 & -0.00071184 & -0.00050222 \\ -0.01597642 & -0.02063644 & -0.01883039 \\ -0.00125061 & -0.00132644 & -0.00351967 \end{pmatrix}$$

Проведените експерименти потвърждават приложимостта на предложената итерационна формула (1.8) (която е еквивалентна на (1.9)) за намиране на стабилизиращо решение на Рикатиевото уравнение (1.5).

Научни приноси в първа глава:

Предложеният итерационен метод чрез формули (1.6) -(1.8) е нов и намира решение на системата нелинейни матрични (1.4). Намереното решение уравнения води до равновение на Наш за линейноквадрична стохастична игра, изследвана от (Zhu, Zhang, 2013). Предложеният итерационен метод е публикуван в (Ivelin Ivanov и V. Tanov, 2018, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, Ann. Acad. Rom. Sci.,2018).

Публикации на автора по първа глава

Ivelin G. Ivanov, **Vladislav Tanov**, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, *Mathematics, and its Applications / Annals of AOSR*, 10(2), 202-210, 2018. (Scopus)

Глава втора. Линейноквадратични игрови модели с двама играчи

Публикуваните приноси на автора във втора глава са в областта на позитивните игри. Отразяваме частта от дисертационния труд, описваща тези научни приноси.

2.4. Игрови модели за позитивни игри

Системи от вида

$$dx = Ax\,dt + B_1u_1dt + B_2u_2\,dt,\ \ x(0) = x_0, \qquad (2.18)$$

се наричат положителни, ако за всички неотрицателни начални състояния x_0 и неотрицателни функции на управление u_1 и u_2, то векторът на състоянието x(t) е неотрицателен във всеки един момент.

Въвеждат се два функционала (i=1,2):

$$J_i(u_1, u_2) = \int_0^\infty (x^T Q_i x + \sum_{j=1}^2 u_j^T\, R_{ij} u_j)\, dt, \text{ за } i = 1,2, \qquad (2.19)$$

Всеки от които трябва да се минимизира, под въздействието на функцията u_i , която е стратегията на i-тия играч.

Ще разгледаме матричното Рикатиево уравнение:

$$0 = -\begin{pmatrix} A^T & 0 \\ 0 & A^T \end{pmatrix}\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} - \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} A - \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} + \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} (S_1 \quad S_2) \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \qquad (2.20)$$

където $(-A)$ е n x n Z-матрица, $S_j = B_j R_{jj}^{-1} B_j^T$ $(S_j = {S_j}^T)$ е неположителна матрица за $j = 1,2$, Q_j – симетрична квадратна неотрицателна матрица с размерност n, R_{jj} – симетрична квадратна отрицателно определена матрица от съответна размерност, B_j – неотрицателна матрица с размерност n x m_j, за $j = 1,2$, а X_1 и X_2 са неизвестните матрици. Ние ще използваме матрици от различни редове.

За управление на положителни системи от горния вид е необходимо да се реши уравнение от вида (2.20). Методът на Нютон е приложен към решаването на уравнение (2.20) и това е направено от Jank, Kremer в (Jank, Kremer, 2005).

Когато записваме $A > 0$ $(A \geq 0)$ за матрица A с размерност n x m ще имаме предвид $a_{ij} > 0$ $(a_{ij} \geq 0)$ за всяко $1 \leq i \leq n$ и $1 \leq j \leq m$, т.е. елементите на матрицата са положителни (неотрицателни). Когато записваме $A > B$ $(A \geq B)$ за матрици A и B с размерност n x m ще имаме предвид $a_{ij} > b_{ij}$ $(a_{ij} \geq b_{ij})$ за всяко $1 \leq i \leq n$ и $1 \leq j \leq m$. За разглежданията матриците Q и S са $Q_k \geq 0$, и

$S_k \leq 0$, k=1,2. В хода на разсъжденията ще използваме факта, че матричното уравнение AXB=C е еквивалентно на линейната система $(B^T \otimes A)\ vec\ X = vec\ C$, където vec означава трансформация на съответната матрица във вектор стълб, следвайки стълбовете на матрицата.

Реална $n\ x\ n$ матрица A ще наричаме Z-матрица, ако съществува реално число s и $C \geq 0\ n\ x\ n$ матрица, т. че $A = sI_n - C$, където I_n е единична матрица от ред n . Реална $n\ x\ n$ неособена матрица $A = (a_{ij})$ се нарича M-матрица, ако $a_{ij} \leq 0$ за $i \neq j$ и $A^{-1} \geq 0$.

Разглеждаме линейните квадратни диференциални игри за положителни линейни системи с обратната структура на информацията и двама играчи. Ускореният метод на Нютон за получаване на стабилизиращото решение на двете уравнения на Riccati е представен в (Jank, Kremer, 2005), където се доказват свойства за сходимост на метода. Освен това, итерационният метод на Lyapunov за изчисляване на равновесната точка Nash е представен в (Baeva, 2016). Освен това се извеждат и доказват свойствата на сходимост на итерационната формула. Реализацията на алгоритъм е илюстрирани на някои числени примери. Известна е следната теорема за свойствата на неотрицателните матрици:

Теорема 2.1. За Z-матрицата A са еквивалентни следните твърдения:

(i) A е M-матрица.

(ii) $A^{-1} \geq 0$.

(iii) $Av > 0$ за всеки вектор $v > 0$.

(iv) Всички собствени стойности на матрицата A имат положителни реални части, т.е. матрицата $-A$ е устойчива матрица.

Акцентът в тази секция за позитивните игри е върху създаването на бързи и ефективни методи за търсене на равновесие по Наш чрез решаване на Рикатиеви уравнения. Такива изследвания сме публикували в следните две наши публикации (Ivan Ivanov, Netov, Tanov, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games, 2016), (Ivelin Ivanov, Tanov, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, 2018).

2.4.1.Метод на Нютон

Методът на Нютон за решаване на уравнението (2.20) е изследван и представен в (Jank, Kremer, 2005) чрез итерационна формула:

$$-K_{i+1}(A - SK_i) - (D - K_iS)K_{i+1} = Q + K_iSK_i \text{ ,}$$

$$i = 0,1,2 \ldots, \qquad (2.21)$$

където $D = \begin{pmatrix} A^T & 0 \\ 0 & A^T \end{pmatrix}$, $Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$, $S = (S_1 \quad S_2)$. Сходимостта на метода е доказана със следващата теорема (Теорема 5 от Jank, Kremer, 2005)

Последният итерационен процес е изсладван за сходимост и за изведени достатъчни условия, гарантиращи тази сходимост (Теорема 2, Ivanov, Netov, Tanov, 2016).

2.4.3.Неявни итерационни формули

Съгласно изследванията, направени от Ma и Lu (Ma, Lu, 2016) тук ще въведем една модификация на метода Нютон разгледан чрез (2.21):

$$-\mathcal{Y}_k(\gamma I + A - SX_k) = (\gamma I_{2n} - D)X_k - \mathcal{Q} \qquad (2.30)$$

$$(\gamma I_{2n} + D - \mathcal{Y}_k S)X_{k+1} = \mathcal{Y}_k(\gamma I - A) - \mathcal{Q}$$

$X_0 = 0$, k=0,1,2, … , $\gamma < 0$.

Последната формула (2.30) наричаме линеаризиран неявен итерационен метод на Нютон и ще използваме съкращението LINI (Linearized Implicit Newton Iteration). Ще представим две твърдения, в които се представят свойства на неотрицателните матрици, като тези свойства ще бъдат полезни в следващите разсъждения.

Записваме матричната функция $\mathcal{R}(\mathcal{X})$ във вида $\mathcal{R}(\mathcal{X}) = \begin{pmatrix} R_1(X_1, X_2) \\ R_2(X_1, X_2) \end{pmatrix}$,

където

$$R_1(X_1, X_2) = -\mathrm{A^T X_1} - \mathrm{X_1 A} + \mathrm{X_1 S_1 X_1} + \mathrm{X_1 S_2 X_2} - \mathrm{Q_1}$$

$$R_2(X_1, X_2) = -\mathrm{A^T X_2} - \mathrm{X_2 A} + \mathrm{X_2 S_1 X_1} + \mathrm{X_2 S_2 X_2} - \mathrm{Q_2} \text{ .}$$

Общото уравнение $\mathcal{R}(\mathcal{X}) = 0$ е еквивалентно на съвкупността от двете уравнения $R_1(X_1, X_2) = 0$ и $R_2(X_1, X_2) = 0$. Може да използваме клетъчната (блочната) структура на матричните коефициенти в (2.30), за да обосновем следващите формули, които задават нов итерационен метод. Ще го наричаме алтернативен линеаризиран неявен разделящ итерационен метод (Alternately

Linearized Implicit Decoupled Iteration (ALIDI)). Дългото име разкрива неговите качества – използва линейни матрични уравнения, неявно достига до решението и всяко итерационно уравнение е независимо от другото:

$$Y_1^{(k)}(\gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T) X_1^{(k)} - Q_1 \qquad (2.31)$$

$$Y_2^{(k)}(\gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T) X_2^{(k)} - Q_2 \qquad (2.32)$$

$$(\gamma I_n + A^T - Y_1^{(k)} S_1) X_1^{(k+1)} = Y_1^{(k)}(\gamma I_n - A + S_2 X_2^{(k)}) - Q_1 \qquad (2.33)$$

$$(\gamma I_n + A^T - Y_2^{(k)} S_2) X_2^{(k+1)} = Y_2^{(k)}(\gamma I_n - A + S_1 X_1^{(k)}) - Q_2 \qquad (2.34)$$

$X_1^{(0)} = X_2^{(0)} = 0$, k=0,1,2, ……, , $\gamma < 0$.

Итерационният метод (2.31) - (2.34) е изведен, изследван и публикуван тук (Ivelin Ivanov, V.Tanov, 2018, стр. 230-244) .

За да изведем свойствата на новите матрични редици, ще отбележим следващите зависимости:

Ще продължим с изследване на предложения итерационен метод (2.31) -(2.34) и извеждане на негови свойства, свързани със сходимостта на метода. Тези свойства ще изведем при някои предположения, които ще формулираме в следващата теорема и като използваме изследванията на Bai и съавтори (Bai и съавтори, 2006).

В следващата теорема, доказана в наша публикация ще изведем достатъчни условия за сходимост на предложения метод.

Теорема 2.4. (Ivelin Ivanov, V.Tanov, 2018, стр. 230-244) Предполагаме, че матрицата (–A) е M-матрица и $Q_1 \geq 0, Q_2 \geq 0$ и $S_1 \leq 0, S_2 \leq 0, \gamma < 0$, така че (- γ I_n − A) е M-матрица и (γ I_n − A) е неположителна. Предполагаме, че съществуват симетрични неотрицателни матрици $\hat{X}_1, \hat{X}_2$, такива че $R_i(\hat{X}_1, \hat{X}_2) \geq 0$, i=1,2 и $-A + S_1 \hat{X}_1 + S_2 \hat{X}_2$ е M-матрица. Построяваме матричните редици $\{X_1^{(k)}\}, \{X_2^{(k)}\}$, $k = 0, \ldots, \infty$, чрез итерационните формули (2.29)-(2.32) удовлетворяват следните свойства:

(i) $\tilde{X}_i \geq X_i^{(k+1)} \geq Y_i^{(k)} \geq X_i^{(k)}$, i=1,2 , k=0,1, ….;

(ii) $R_i\left(X_1^{(k)}, X_2^{(k)}\right) \leq 0$, $R_i\left(Y_1^{(k)}, Y_2^{(k)}\right) \leq 0$, $R_i\left(X_1^{(k+1)}, X_2^{(k+1)}\right) \leq 0$, i=1,2, k=0,1, ….;

(iii) Матричните редици $\left\{X_1^{(k)}\right\}, \left\{X_2^{(k)}\right\}$, $k = 0, \ldots, \infty$, са сходящи към минималното неотрицателно решение $\tilde{X}_1, \tilde{X}_2$ на двойката Рикатиеви уравнения $R_1(X_1, X_2) = 0$ и $R_2(X_1, X_2) = 0$, за които $\tilde{X}_i \leq \hat{X}_i$ и матрицата $\mathrm{A} - \mathrm{S}_1 \tilde{X}_1 - \mathrm{S}_2 \tilde{X}_2$ е устойчива.

Теоремата е представена без доказателство.

Числени примери

Ще приложим описаните итерационни методи за намиране на равновесието върху два примера с конкретни данни.

Пример 2.5. Матричните коефициенти A, Bi, Qi и Rii за i=1,2 описваме в средата на Matlab.

```
A=abs(randn(n))/99;   s=max(abs(eig(A)))+4.5;  \gamma= -5.0;
for i=1:n,   A(i,i)=-(A(i,i))-s;   end
B1  = abs(randn(n,1))/2;
B2 = eye(n,n);      B2(n,n)=n/3;
Q1=zeros(n,n);   Q1(1,1)=n/2;   Q1(n,n)=1.5;
Q2 = 2 Q1;    R11=-1;   R22 = -eye(n,n);
R22(1,1)=-50;     R_{22}(n,n) = -30;
```

Изпълняваме Пример 2.5 за различни стойности на n, и по 100 повторения за вяска стойност на n. Избираме $X_1^{(0)} = X_2^{(0)} = 0$, и установяваме $R_i\left(X_1^{(0)}, X_2^{(0)}\right) = -Q_i \leq 0$, т.е. матрицата е неположителна. Избираме nx n матриците

$$\hat{X}_1 = \begin{pmatrix} 0.3 & 0.01 & \cdots & 0.01 & 0.01 \\ 0.01 & 0.3 & \cdots & 0.01 & 0.01 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0.01 & 0.01 & \cdots & 0.3 & 0.01 \\ 0.01 & 0.01 & \cdots & 0.01 & 0.3 \end{pmatrix}$$

$$\hat{X}_2 = \begin{pmatrix} 0.5 & 0.01 & \cdots & 0.01 & 0.01 \\ 0.01 & 0.5 & \cdots & 0.01 & 0.01 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0.01 & 0.01 & \cdots & 0.5 & 0.01 \\ 0.01 & 0.01 & \cdots & 0.01 & 0.5 \end{pmatrix}$$

Условията на Лема 2.5 и Лема 2.6 и Теорема 2.4 са изпълнени за матриците $\hat{X}_1, \hat{X}_2$, т.е. $\hat{X}_1 \geq X_1^{(0)}, \hat{X}_2 \geq X_2^{(0)}$, $R_i\left(X_1^{(0)}, X_2^{(0)}\right) \leq 0$, $R_i\left(\hat{X}_1, \hat{X}_2\right) \geq 0$, i=1,2. Матрицата $(-\mathrm{A} + \mathrm{S}_1 \hat{X}_1 + \mathrm{S}_2 \hat{X}_2)$ е неособена М-матрица. Пресметнатото решение $\tilde{X}_1, \tilde{X}_2$ удовлетворява следните неравенства $\tilde{X}_1 \leq \hat{X}_1, \tilde{X}_2 \leq \hat{X}_2$ и

допълнително $(-\mathrm{A}+\ \mathrm{S}_1\ \tilde{X}_1, +\ \mathrm{S}_2\ \ \tilde{X}_2)$ е M-матрица. В Таблица 2.2. са описани резултатите от експериментите. Предложеният неявен метод чрез формули (2.31) -(2.34) е по-бърз независимо, че прави повече среден брой итерационни стъпки.

Пример 2.6. Матричните коефициенти определяме:

```
A=abs(randn(n))/10;        s=max(abs(eig(A)))+4.5;  \gamma= -5.0;
for  i=1:n,   A(i,i)=-(A(i,i))-s;    end
B1 = abs(randn(n,1))/2;
B2 = eye(n,n);    B2(n,n)=abs(randn);
Q1=zeros(n,n);      Q1(1,1)=n/2;    Q1(n,n)=1.5;
Q2 = 2 Q1;            R11=-1;
R22 = -eye(n,n);      R22(1,1)=-80;    R22(n,n)=-90;
```

В таблица 2.2 и 2.3 са представени резултатите за различни стойности на n. Направени са по 100 повторения за конкретна стойност на n. Отразени са максималния рой итерационни стъпки за достигане до решението (maxIt), среден брой итерационни стъпки (avIt) и процесорното време за изпълнение на всички повторения (CPU) . Описани са резулататите за метода на Нютон (2.21) и метода ALIDI (2.31)-(2.34).

Таблица 2.2.

	Метод на Нютон NI (2.21)			ALIDI (2.31)-(2.34)		
N	maxIt	avIt	CPU	maxIt	avIt	CPU
8	4	3.1	0,31сек	6	5,01	0,125сек
16	4	3,27	0,43сек	10	5,5	0,19сек
24	5	3,57	0,73сек	17	6,67	0,37сек
32	5	3,67	1,16сек	14	8,06	0,73сек

Таблица 2.3.

	Метод на Нютон NI (2.21)			ALIDI (2.31)-(2.34)		
N	maxIt	avIt	CPU	maxIt	avIt	CPU
15	4	3,2	0,39сек	8	5,11	0,1сек
25	4	3,38	0,73сек	12	6,9	0,29сек
40	6	3,69	1,66сек	10	8,85	0,89сек
55	6	3,66	3,25сек	22	10,8	2,03сек

2.5.Подобрен итерационен метод

В следваща наша публикация (Ivelin Ivanov, Vladislav Tanov, 2020) подобряваме итерационната формула (2.31) -(2.34) и предлагаме нова модификация

$$Y_1^{(k)}(\gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T) X_1^{(k)} - Q_1 \qquad (2.35)$$

$$Y_2^{(k)}(\gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T) X_2^{(k)} - Q_2 \qquad (2.36)$$

$$(\gamma I_n + A^T) X_1^{(k+1)} = Y_1^{(k)}(\gamma I_n - A + S_1 Y_1^{(k)} + S_2 X_2^{(k)}) - Q_1 \qquad (2.37)$$

$$(\gamma I_n + A^T) X_2^{(k+1)} = Y_2^{(k)}(\gamma I_n - A + S_1 X_1^{(k)} + S_2 X_2^{(k)}) - Q_2 \qquad (2.38)$$

$X_1^{(0)} = X_2^{(0)} = 0$, k=0,1,2, ……, , $\gamma < 0$.

За итерационния процес (2.35) -(2.38) отбелязваме следните свойства:

Теорема 2.5. (Ivelin Ivanov, Vladislav Tanov, 2020) Предполагаме, че матрицата (–A) е M-матрица и $Q_1 \geq 0, Q_2 \geq 0$ и $S_1 \leq 0, S_2 \leq 0, \gamma < 0$, така че ($\gamma I_n - A$) е M-матрица и ($\gamma I_n - A$) е неположителна. Предполагаме, че съществуват симетрични неотрицателни матрици $\hat{X}_1, \hat{X}_2$, такива че $R_i(\hat{X}_1, \hat{X}_2) \geq 0$, i=1,2 и $-A + S_1 \hat{X}_1 + S_2 \hat{X}_2$ е M-матрица. Построяваме матричните редици $\{X_1^{(k)}\}, \{X_2^{(k)}\}$, $k = 0, \ldots, \infty$, чрез итерационните формули (2.35) -(2.38). Матричните редици удовлетворяват следните свойства:

(i) $\tilde{X}_i \geq X_i^{(k+1)} \geq Y_i^{(k)} \geq X_i^{(k)}$, i=1,2 , k=0,1, ….;

(ii) $R_i(X_1^{(k)}, X_2^{(k)}) \leq 0$, $R_i(Y_1^{(k)}, Y_2^{(k)}) \leq 0$, $R_i(X_1^{(k+1)}, X_2^{(k+1)}) \leq 0$, i=1,2, k=0,1, ….;

(iii) Матричните редици $\{X_1^{(k)}\}, \{X_2^{(k)}\}$, k=0,…,∞, са сходящи към минималното неотрицателно решение $\tilde{X}_1, \tilde{X}_2$ на двойката Рикатиеви уравнения $R_1(X_1, X_2) = 0$ и $R_2(X_1, X_2) = 0$, за които $\tilde{X}_1 \leq \hat{X}_i$.

(iv) Ако $-A + S_1 \hat{X}_1 + S_2 \hat{X}_2$ е M-матрица и лллл, решението $\tilde{X}_1, \tilde{X}_2$ е ляводясно стабилизиращо решение двойката Рикатиеви уравнения $R_1(X_1, X_2) = 0$ и $R_2(X_1, X_2) = 0$.

Доказателството следва подобни разсъждения на лема 2.8.

Пример 2.7. Матричните коефициенти определяме: (в термините на Matlab)

```
A=[-2.74 0.06 0.015 0.099;
   0.2 -2.5 0.064 0.08;
   0.004 0.15 -2.56 0.09;
   0.14 0.12 0.21 -2.57];

B1=[0.5938; 0.2985; 0.49; 0.98];
B2=[2.8  0   0  0;
      0  2.9  0  0;
      0   0  2.84 1.5;
      0   0  1.5 1.3];
```

Q1=eye(n,n)/2; Q1(1,1)=n/2; Q1(n,n)=1.5;

Q2 =.5 * Q1;

R11=-1.909;

R22 = -eye(n,n); R22(1,1)=-50; R22(n,n)=-30;

S1=B1*inv(R11)*B1';

S2=B2*inv(R22)*B2';

Резултатите са представени в Таблица 2.4.

Таблица 2.4.

	ALIDI (2.31) -(2.34)		(2.35) -(2.38)	
γ	avIt	CPU	avIt	CPU
-5	402	2,67сек	431	2,7сек
-3	256	1,76сек	278	1,78сек
-1	112	0,82сек	112	0,79сек
-0.5	40	0,37сек	39	0,34сек
-0.25	80	0,62сек	77	0,57 сек

Приноси във втора глава:

Приносите във втора глава са предложените два нови итерационни метода за намиране на решение на клетъчно Рикатиево уравнение със специални коефициенти:

итерационен метод (2.31) -(2.34),

итерационен метод (2.35) -(2.38).

За двата итерационни метода теоретично са изведени свойствата за тяхната сходимост. И двата метода се отличават с ясна схема за изпълнение и лесната компютърна реализация. Експериментите демонстрират тяхната ефективност.

Публикации на автора по втора глава.

1.Ivan Ivanov, Nikolay Netov, Vladislav **Tanov**, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games. International Journal of Mathematical and Computational Methods, **1**, 378-381, 2016

2.Ivelin Ivanov, Vladislav **Tanov**, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, Mathematics and its Applications / Annals of AOSR, 10(2), 230-244, 2018. (Scopus)

3.Ivelin Ivanov, Vladislav **Tanov**, A Nonsymmetric Nash-Riccati Equation and Decoupled Schemes for a Stabilizing Solution, Applied Mathematics E-Notes, 20(2020), 357-366, Applied Mathematics E-Notes, 20(2020), 357-366. (Scopus)

Глава трета. Оптимизация на небалансирани множества

Методи и алгоритми

Небалансираните множества, още познати като класов дисбаланс, са често срещани в областта на „машинно самообучение“ (Samuel, 1959) с приложение в различни области, например откриване на сърдечно-съдови и чернодробни заболявания, нефтени разливи в сателитни изображения, както и задачи за извличане и филтриране на информация, и други (Raskutti and Kowalczyk, 2004) (Wu and Chang, 2003). Задачата за подобряване на ефективността на проблемите с класовият дисбаланс, състоящи се от клас на мнозинството (по-голям брой наблюдения) и клас на малцинството, е много важен аспект за различни оптимизационни проблеми. Целта е чрез оптимизация на класа на мнозинството да се достигне до оптимизирано, балансирано подмножество, което да се използва за прогнозиране с високи стойности на точност.

Bohacik и Zabovsky са проучили вероятностна реализация с дадена контролирана дискретизация, използвайки експертни познания в областта на сърдечните заболявания (Bohacik and Zabovsky, 2019). Алгоритмичната методология е приложена в среда на Waikato за анализ на знания като клас NaïveBayes с дискретизацията на числовите атрибути на Fayyad-Irani (Fayyad and K. B. Irani, 1993). Изследването се основава на k-кратно (k=10) кръстосано валидиране и използва чувствителност, специфичност и тяхната сума като мерки. Чувствителността (истински положителни проценти, *sensitivity*) представлява способността на алгоритъма да идентифицира истински положителни (TP) случаи по отношение на всички положителни резултати, със следния израз: TP/(TP+FN). Фалшивите отрицателни (FN) ще се считат за отрицателни случаи, докато всъщност са положителни. Специфичността, *specificity*, посочва способността на алгоритъма да идентифицира случаи на истински отрицателни нива (TN) по отношение на всички отрицателни резултати, както следва: TN/(TN+FP) . Фалшивите положителни (FP) ще се считат за положителни случаи, докато всъщност са отрицателни. Bohacik и Zabovsky използват сумата от чувствителност и специфичност като цялостна точкова система за сравнение на алгоритми (Bohacik and Zabovsky, 2019).

Експериментите, които провеждаме, са върху няколко множества от данни, използвани от други автори и свободно достъпни в интернет, показани в Таблица 3.1. Резултатите от експериментите показват предимство пред разгледаните по-горе методи за работа с данни, като повторно семплиране (*resampling*), бутстрап (*bootstrap*) и подбор на статистически значимите променливи (*feature selection*). Ще разглеждаме следната хипотеза: *Алгоритъмът за оптимизация на небалансирани множества е ефективен за подобряване на точността при прогнозиране с касификационни модели*. Хипотезата се проверява към конкретно множество при получените мерки за стандартно оценяване на класификцаионни модели, като точност (Accuracy), прецизност (Precision), рикол (Recall) и f_1 показател (F_1 Score)

Таблица 3.1

Множества	Параметри	Класове - Наблюдения	Източник
Diabetes	8	Клас 0 – 500 Клас 1 - 268	уебстраница
Statlog Heart	13	Клас 1 – 150 Клас 2 - 120	уебстраница
Indian Liver Patient Dataset (ILPD)	10	Клас 1 – 416 Клас 2 - 167	уебстраница

В тази глава ще формулираме оптимизационна задача за провеждане на класификационен анализ на големи данни.

В следващите секции ще разгледаме алгоритъм за решаване на оптимизационната задача, т.е. алгоритъм за оптимизация на небалансирани множества, и ще го сравним с резлултатите постигнати от дугите автори, като използваме същите множества и приложените от тях класифициращи алгоритми – RF, SVM, KNN, DT, NB и LR. Резултатите от изследването и проведеният анализ са публикувани в Data Centric Optimization Method to Imbalanced Datasets, като част от *International Conference on Mathematical and Statistical Physics, Computational Science, Education, and Communication* (Vladislav Tanov, Ivan Ivanov, 2023) и статията Data-Centric Optimization Approach for Small, Imbalanced Datasets, публикувана в *Journal of Information and Organizational Sciences (JIOS)* (Vladislav Tanov, 2023).

3.1. Модел и алгоритъм за оптимизация на небалансирани двоични множества (Data Centric Optimization - DCO)

Формулираме следния модел. Знаем множество от данни X, в което са известни наблюденията от различните класове. Разделяме на две множества Xtrain и Xtets в отношение Xtrain : Xtets = 80:20 или Xtrain : Xtets = 70:30. При това деление от наблюдения от всеки клас попадат и в двете множества. Множеството Xtrain ще използваме за построяване на класификационне модел , докато върху Xtets ще се проверяват оценъчните индикатори за модела. Избираме модел за класификационен анализ със стойности на съответните входни параметри.

Да се максимизира функцията Acc(Xtest)

при условия

1.Построяване по подходящ начин на множеството Xtrain.

2.Избор на параметрите на класификационния модел.

Функцията Acc(Xtest) измерва точността на класификационния анализ, след като са определени Xtrain и Xtets, моделът е построен върху Xtrain и общата точност (accuracy) е пресметната върху Xtest по формулата

$$\mathrm{Acc(Xtest)} = \frac{\sum_{i=1}^{k} cm_{ii}}{\sum_{i,j=1}^{k} cm_{ij}} ,$$

където $CM = (cm_{ij})$ е confussion matrix , съдържаща k-класа от наблюденията.

Решението на всяка оптимизационна задача се търси по емпиричен път (поне засега), в зависимост от техниките за деление на дадено множество на две множества и възможностите за избор на параметрите на избран класификационен модел.

Тук ще предложим алгоритъм DCO за решаване на дефинираната оптимизационна задача (Vladislav Tanov, 2023).

Основната задача на DCO е да изследва и раздели небалансираното двоично множеството, или множество с двоичен класов дисбаланс, на балансирано подмножество, използвайки *undersampling* или т.нар. разбъркване на извадка. Тази идея следва примера за продобиване на инфорамция, *information gain,* разгледан от Shaltout и съавтори. Те използват *information gain* като методология за селектиране на статистически значими променливи базирана на ентропичната величина като мярка за безпорядък (Shaltout и съавтори, 2020). За целта, алгоритъмът за оптимизация на небалансирани множества запазва цялостта на класа на малцинството, *minority class*, и селектира оптимизирано подмножество от класа на мнозинството, така че приложеният класификационен алгоритъм достига най-високи стойностти на точност. Оптимизацията следва логиката на минимизиране на грешките които калсификициония модела допуска при валидацията на прогнозираните стойностти, наречен *model error rate (mer)*. С други думи, DCO филтрира така наречените „bad“ или „noisy” редове от подмножеството на класа на мнозинството.

Следвайки класическия подход, балансираното подмножество бива разделено на тренировъчно и тестово със съотношение от 80 към 20, респективно. Процесът продължава докато *mer* достигне 0 или се изразходят броят на случайините положителни величини (между 0 и 100) при селектирането на подмножества от класа на мнозинството. Експериментите с DCO са правени със следните компютърни храктеристики: (RAM: 16GM, CPU: 2.6GHz 6-Core Intel Core i7), както следва:

Декларираме следните параметри:

- $i \in \{0 \ldots 100\}$
- r_i – random under-sampling integer
- n – length of the given dataset
- m – minority class length
- R – list of integers
- D_n – given dataset
- X_n – random variable (# of variables in D_n)
- Y_n – response variable (# of classes in D_n), Y = 1,…K, where K>=2
- $D_{i,m}$ – balanced, under-sampled data sub-set:
  $D_{i,m} = ([X_1,Y_1],\ldots[X_n,Yn] \mid r_i , m)$, where [X,Y] is independent of D_n
- *mer* - model error rate, mer = 100
- T_o and V_o – optimized train and validation sub-sets

ALGORITHM: *DCO OPTIMIZATION PHASE*

Initialization of variables listed above.
set ***optimized*** *= False*
while *not* ***optimized***
    draw random integer – r_i
    if random integer (r_i*) not in list of integers (R)*
        append random integer (r_i*) to R*
        $\boldsymbol{D_{i,m}}$ *= undersample(*D_n *|* r_i *, m)*
        split $D_{i,m}$ *into train and validation sets (80/20):*
        $\boldsymbol{T_i, V_i}$ *= train_test_split (*$D_{i,m}$ *| .20)*
        $\boldsymbol{C_i}$ *= build classifier*
        fit train set to C_i *and calculate false positive error (FPE) and false negative error (FNE). Keep track of* C_i *error:*
        $\boldsymbol{error_{Ci}} = C_i\ (T_i, V_i) = \Sigma\ [(FPE_i \mid C_i, D_{i,m}),(FNE_i \mid C_i, D_{i,m})]$
        evaluate current model error rate (mer)
        if ***mer*** *is greater than* $error_{Ci}$
            mer *=* $error_{Ci}$
            $\boldsymbol{T_o} = T_i$
            $\boldsymbol{V_o} = V_i$
    if lenght of R is greater than 100: ***optimized*** *= True*
end

3.1.1. Бутстрап (*bootstrap*) процедурате при класифициращи модели

Диабетът е много често срещано заболяване, при което се изисква постоянно наблюдение и контролиране, за да се избегнат фатални последици. Azberbg и съавтори правят експерименти за прогнозиране на пациенти с диабет прилагайки алгортимичният метод на „случайно избраните гори“ (Random Forest, RF). RF има широко пролижение и е смятан за един он стандартните методи при използването на контролирани модели, *suppervised machine learning*, Техният експеримент агрегира колекция от модели конструирани и тренирани въз основа на бутстрап процедурата за разделяне на данните на тренировъчно и тестово подмножество, с използването на замяна, или т.нар. *bootstrap samples with replacemetn*. Всяка итерация е базирана на случаен подбор на променливи, които изграждат ротационна матрица, с комбинации от променливи. Това води то генериране на различни вариации от калсификациращи модели, дефинирани в уравнение 3.4 (Azbeg и съавтори, 2022).

Таблица 3.1.1-1

Множества	Класове - Наблюдения	ACA точност на RF(%)	DCO точност на RF(%)
Diabetes 1	Клас 0 – 500 Клас 1 – 268	78.65	**87.04**
Diabetes 2	Клас 0 – 1316 Клас 1 – 684	99.5	**99.65**
Diabetes 3 (1 и 2)	Клас 0 – 1816 Клас 1 – 952	99.8	**100**

В таблици 3.1.1-1 и 3.1.1-2 са представени и сравнени резултатите, получени от Azberbg и съавтори от техния алгоритъм (ACA) с резултатите, полчени от експериментите с прилагането на алгоритъма за оптимизация на небалансирани множества (Data Centric Optimization - DCO).

Експериментът, направен от Azberbg и съавтори използва алгоритмичен (ACA) метод като включва сравняване на получените резултати при тестване с множеството *Diabetes1* със следните класифициращи алгоритми за машинно самообучение, като модела с поддържащи вектори (Support Vector Machines, SVM), модела на на-близките съседи (K-Nearest Neighbor, KNN), модела на дърветата на решенията (Decision Tree-baded, DT), модела на адаптивно усилване (Adaptive Bosting, ADABoost), модела на изкуствените невронни мрежи (Artificial Nearal Network, ANA) и модела на логистичната регресия (Logistic Regression, LR). Таблица 3.1.0 ще представим и сравним резултатите получени от SVM, KNN, DT, ADABoost, ANN, LR, DL и ACA (RF) с предложеният от нас алгоритъмът за оптимизация на небалансирани множества (Data Centric Optimization - DCO) приложени върху множеството *Diabetes1*.

Таблица 3.1.1-2

Автори	Публикация	Алгоритъм	Точност % (Accuracy)
Wei и съавтори	2010 г.	SVM	73
Panwar и съавтори	2016 г.	KNN	78
Ramezankhani и съавтори	2016 г.	DT	74
Mingqi, Xiaoyang и Dongdong	2020 г.	ADABoost	79.2
Pradhan и съавтори	2020 г.	ANN	80.4
Tigga и Garg	2021 г.	LR	75.32
Ihnaini и съавтори	2021 г.	DL	72.7
Azbeg и съавтори, 2022 (ACA)	2022 г.	RF	85.9
DCO	2023г.	RF	87.04

Azberbg и съавтори правят допълнителни експерименти с множествата *Diabetes2,* и комбинация от *Diabetes1* и *Diabetes2* което наричат *Diabetes3 (Diabetes1 + Diabetes2)*. По този начин авторите създават множества с по-голям брой наблюдения, което е стандартно прието за увеличаване точността на разглежданите модели (Bailly и съавтори, 2022).

Както се вижда в горната таблица, DCO дава по-добри резултати от приложените методи в експериментите на Azberbg и съавтори (ACA). Както би се очаквало, точността на моделите се увеличава с увеличението на наблюденията в множествата, което всъщност е и главната идея в експеримента на Azberbg и съавтори. Тук се поставя въпроса, при по-задълбочено оценяване на резултатите, фактически колко са „добри“ тези два подхода.

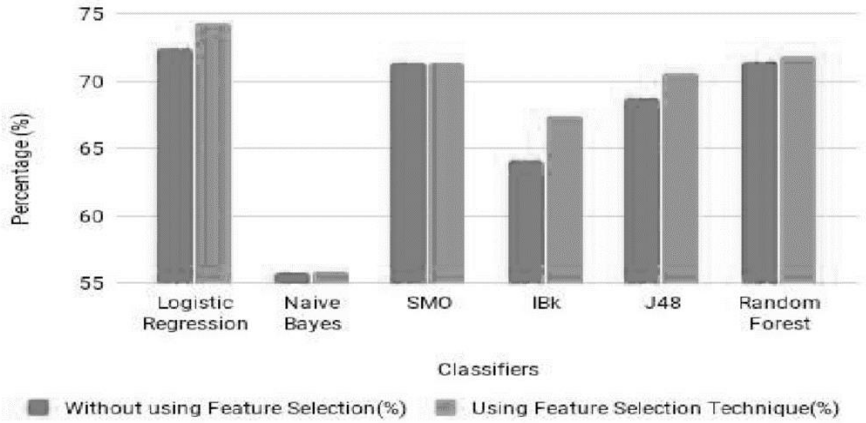
Когато разгледаме и анализираме Фигура 3.1.1 с матрицата с допуснати грешки от моделите, *confusion matrix*, забелязваме че високата точност на метода на Azberbg и съавтори (ACA), при прогнизиране на Diabetes 1 множество, се дължи на по-големия процент вярно прогнозирани стойности когато пациентите нямат диабет, или специфичността, като прави грешки само 15% (19 от 125) от времето. Но когато даден пациент всъщност има диабет, или чуствителността на метода да прогнозира случаи с диабет, техният метод прави грешки ~33% от времето (22 от 67). Това означава, че техният метод ще назначава лечение от диабет на всеки 33 пациенти от 100, които посещават клиничното заведение, т.е. този метод всъщност не успява да прогнозира достатъчно добре.

Това се дължи на неблансираното множество, което дава предпоставки за конструиране на предубедени модели, *biased models*, като в този случай методът на Azberbg и съавтори е предубеден към пациенти с диабет. В същото време, DCO дава по-добри резултати, като *специфичността* (*specificity*) на метода е 87%, т.е. прави грешки 13% (7 от 54) от времето. В същото време, когато пациентите реално имат диабет, DCO прави грешки само 9% (5 от 54) от времето, което е почти четири (3.66) пъти по-малко сгрешени даигнози в сравнение с метода на Azberbg и съавтори.

3.1.2. Подбор на статистически променливи (feature selection)

Singh и съавтори експериментрат с данни за Индийски пациенти с чернидробни заболявание (Indian Liver Patient dataset, ILPD), които са публично достъпни в базата от данни на Калифорнийският Университет в Ървайн, Калифорния (UCI Machine Learning Repository: ILPD - Indian Liver Patient Dataset). ILPD множеството съдържа 416 наблюдения с пациенти с чернодробно заболяване (Клас 1) и 167 (Клас 2) наблюдения с пациенти без чернодробно заболяване (UCI Machine Learning Repository). Върху ILPD множеството са приложени две методологии за подбор на статистически променливи заедно с 10-кратното кръстосано валидиране, *10-fold cross-validation*, а именно корелация базирана между променливите, *correlation-based feature selection*, която селектира променливите, чрез добавяне или пермахване, докато не се достигне до спад в оценката. Singh и съавтори използват Waikato Environment for Knowledge Analysis, WEKA (Singh, Bagga, Kaur, 2020), който е се смята за един от най-практичните публично достъпни софтуеъри за анализиране на данни (Written, Frank, Hall, 2022).

Фигура 3.1.2



Както може да се види в Таблица 3.1.2-1 с и без прилагането на подбор на статистически променливи, логистичната регресия (LR) дава най-високи резултати с и без прилагането на подбор на статистически променливи, като точността, *accuracy*, на LR е 74.36% и 72.5% респективно. Алгоритъмът на „случайно избраните гори“ (Random Forest, RF) достига по-ниски стойности на точност от 71,87% и 71.53%, респективно (Singh, Bagga, Kaur, 2020).

Таблица 3.1.2-1

Множество	Класове – Наблюдения	Singh and co точност на LR (%)		Singh and co точност на RF(%)		DCO точност на RF(%)
		No FS	FS	No FS	FS	
Indian Liver Patient dataset (ILPD)	Class 1 – 416 Class 2 – 167	72.5	74.4	71.53	71.9	**92.54**

Това показва, че приложената методология за подбор на статистически променливи достига по-слаби резултати в сравнение с предложения от нас алгоритъм за оптимизация на небалансирани множества (Data Centric Optimization - DCO). DCO в комбинация с RF достига до много по-висока точност от 92.54%, както се вижда в Таблица 3.1.2-1.

В Таблица 3.1.2-2 ние даваме допълнителните тестови мерки за цялостно оценяване на модели, а именно прецизност (Precision), рикол (Recall), f_1 показател (F_1 Score, F1) както и площ под кривата (Area Under Curve, AUC).

Таблица 3.1.2-2

Множество: Indian Liver Patient dataset (ILPD)	DCO тестови мерки AUC = 92.5			
	Accuracy	Precision	Recall	F1
Клас 1 (34)	92.54	91.43	94.12	92.75
Клас 2 (33)	92.54	93.75	90.91	92.31

3.1.3. Вероятностна реализация с дадена контролирана дискретизация

Bohaick и Zabovsky експериментират с вероятностен подход за подпомагане на вземането на решения за диагностика на сърдечни заболявания (Bohaick, Zabovsky, 2019). Центърсj за контрол и превенция на заболяванията (Centers of Disease Control and Prevention, CDC) свързва термина „сърдечно заболяване“, *heart disease*, с няколко типа сърдечни заболявания (най-често срещаното е коронарната артериална болест), което е водещата причина за смърт в Съединените щати (CDC, 2022). За целта, Bohaick и Zabovsky използват Statlog Heart множеството, което е публично достъпно в базата от данни на Калифорнийският Университет в Ървайн. Statlog Heart множеството се състои от 120 наблюдения (Клас 2) диагностицирани със сърдечно заболяване и 150 наблюдения (Клас 1) без диагностицирано сърдечно заболяване (UCI Machine Learning Repository: Statlog Heart).

Резултатите от експерименти с предложения от Bohaick и Zabovsky алгоритъм са сравнени с няколко алгоритми за машинно самообучение като NB, изкуствени невронни мрежи (NN), изкуствени невронни мрежи с многослоен перцептрон (MLP) и модела на дърветата на решенията (Decision Tree-baded, DT). Те използват сумарната стойност от чуствителността и специфичността на тестовите модели като мярка за сравнение (Bohaick, Zabovsky, 2019). В Таблица 3.1.3-1, ние сравняваме техните публикувани резултати, с резултатите получени от представения от нас алгоритъм за оптимизация на небалансирани множества (Data Centric Optimization - DCO).

Както се вижда от Таблица 3.1.3-1, методологията на Bohaick и Zabovsky постига най-високи резултати на чувствителност и специфичност от 0.90 и 0.842 (сумарно 1.742), съответно, в сравнение с другите алгоритми за машинно самообучение.

Таблица 3.1.3-1

Множество: Statlog Heart Алгоритми	Чуствителност (Sensitivity)	Специфичност (Specificity)	Сумарен сбор (Sum)
NB-Mod	0.900	0.842	1.742
NB	0.840	0.817	1.657
MLP	0.880	0.800	1.680
DT	0.840	0.692	1.532
NN	0.773	0.717	1.490
DCO	**0.96**	**1.00**	**1.96**

Предложеният от нас алгоритъм за оптимизация на небалансирани множества (DCO) превъзхожда значително техният, достигайки тестова чувствителност и специфичност от 0.96 и 1.0 (сумарно 1.96). В допълнение на тестови мерки за цялостно оценяване на модели, а именно прецизност (Precision), рикол (Recall), f_1 показател (F_1 Score, F1) както и площ под кривата (Area Under

Curve, AUC), в Таблица 3.1.3-2 ние отчитаме следните резултати, получени от DCO.

Таблица 3.1.3-2

Множество: Statlog Heart	DCO тестови мерки AUC = 97.9			
	Accuracy	Precision	Recall	F1
Клас 1 (24)	97.9	1.00	95.83	97.87
Клас 2 (24)	97.9	96.0	1.00	97.96

3.1.4. Оптимизационнен метод на рояка частици за подбор на статистически значимите променливи

Dubey, Sinhal и Sharma разрабботват експерименти използвайки оптимизационен метод на рояка частици за подбор, за да се достигне до оптимален набор от категориални променливи, или още познат като Improved Auto Categorical Particle Swarm Optimization (IACPSO). Чрез оптимзиация на рояка частици (Particle Swarm Optimization, PSO) се автоматизира подхода за избор на оптималните стойности на контолираните параметри при всяка итерация. Dubey, Sinhal и Sharma анализират поведението и въздействието на оптималните числени параметри вътху различни алгоритми за машинно самообучение (Dubey, Sinhal и Sharma, 2022).

Dubey, Sinhal и Sharma предприемат необичаен подход за разделяне на множеството на тренировъчно и тестово подмножество, като използват съотношение от 75/25. За да осигурим правилно сравнение между техния метод, IACPSO, и представеният от нас алгоритъм за оптимизация на небалансирани множества (Data Centric Optimization - DCO), ние проведохме експерименти, използвайки както стандартното сътоношение от 80/20 за разделяне на тренировъчно и тестово подмножество, така и 75/25, приложено върху същите множества Statlog, Cleveland и Hungarian. В следващата Таблица 3.1.4 са изложени резултатите и сравненията между двете методологии.

Таблица 3.1.4

	Cleveland					**Statlog**					**Hungarian**				
MLA	AC	PR	SV	FS	MCC	AC	PR	SV	FS	MCC	AC	PR	SV	FS	MCC
LR	87	87	87	87	74	91	92	89	90	81	88	88	88	88	77
LR + IACSPO	**96**	**97**	97	**97**	89	**98**	**99**	97	**99**	95	97	**98**	98	**98**	92
LR - DCO (75/25)	94	92	97	94	89	**98**	97	**100**	98	**97**	94	93	96	95	89
LR - DCO (80/20)	**96**	93	**100**	**97**	**93**	**98**	96	**100**	98	96	**98**	95	**100**	**98**	**95**
DT	82	83	83	82	65	83	85	81	82	65	82	83	82	82	65
DT + IACSPO	92	**93**	94	93	83	93	**96**	94	**96**	83	93	93	94	93	92

DT - DCO (75/25)	93	91	94	93	86	95	94	97	95	90	96	**96**	96	96	92
DT - DCO (80/20)	**96**	**93**	**100**	**97**	**93**	**96**	92	**100**	**96**	**92**	**98**	**96**	**100**	**98**	**95**
RF	87	87	87	87	71	89	90	87	88	61	86	87	86	87	69
RF + IACSPO	97	97	**98**	97	90	97	**98**	98	**98**	89	97	98	**98**	**98**	88
RF - DCO (75/25)	96	94	97	96	91	97	97	97	97	93	92	96	89	92	85
RF - DCO (80/20)	**99**	**100**	97	**99**	**97**	**98**	97	**100**	**98**	**97**	**98**	**100**	96	**98**	**96**
SVM	87	87	87	87	74	87	87	86	87	73	87	87	87	87	74
SVM + IACSPO	97	97	98	97	90	97	98	98	**98**	89	97	98	98	98	88
SVMGS	89	89	89	89	77	89	93	86	88	78	89	89	89	89	77
SVMGS + IACSPO	98	97	96	97	90	**98**	**99**	98	97	90	**97**	**97**	97	**97**	90
SVM - DCO (75/25)	94	94	94	94	88	97	94	**100**	97	94	94	93	96	95	89
SVM - DCO (80/20)	**100**	**100**	**100**	**100**	**100**	**98**	96	**100**	**98**	**96**	95	91	**100**	95	**91**
KNN	76	76	75	76	51	76	73	75	75	50	77	76	76	76	53
KNN + IACSPO	92	92	91	92	84	92	92	93	90	81	91	91	91	89	80
KNN - DCO (75/25)	94	94	94	94	88	**97**	94	**100**	**97**	**94**	94	93	96	94	89
KNN - DCO (80/20)	**96**	**96**	**96**	**96**	**93**	96	**96**	96	96	92	**98**	**96**	**100**	**98**	**95**
NB	87	87	87	87	74	91	92	89	90	81	88	88	88	88	76
NB + IACSPO	92	93	92	91	83	94	95	94	92	85	92	92	92	91	82
NB- DCO (75/25)	93	91	94	93	86	97	94	100	97	94	**100**	**100**	**100**	**100**	**100**
NB- DCO (80/20)	**100**	**100**	**100**	**100**	**100**	**98**	**96**	**100**	**98**	**96**	**100**	**100**	**100**	**100**	**100**

Анализирайки Таблица 3.1.4, можем да видим общо 90 мерки за трите множества от данни, Statlog, Cleveland и Hungarian, които включват 5 показателя, точност (Accuracy, AC), прецизност (Precision, PR), чуствителност (Sensitivity, SV), f_1 показател (F_1 Score, FS) и коефициент на корелация на Matthew (MCC). Това се равнява на 15 мерки за всеки приложен алгоритъм за машинно самообучение, *6 MLA*, (3 множества, умножени по 30 - 5 показателя за всеки алгоритъм). Като разгледаме по-отблизо изброените мерки, можем да забележим, че предложеният от нас алгоритъм за оптимизация на небалансирани множества (Data Centric Optimization - DCO) превъзхожда IACPSO метода 75.56% (68/90) процента от времето със средно 5.62 точки за всяка мярка. С малки изключения за LR, DT и SVM, където 13.33% (12/90) от времето IACPSO дава по-добри резултати, и 11.11% (10/90) от времето, когато DCO и IACPSO дават същите резултати.

3.2. Алгоритъм за оптимизация на небалансирани множества с мулти класове, повече от два класа, (Data Centric Multiclass Optimization - DCMO).

Класификацията на небалансирани данни, класов дисбаланс, се проявява в два аспекат. Първият случай е когато имаме наличие на два класа с отрицателен и положителен класови етикет, или двоичен класов дисбаланс. Във вторият случай имаме повече от един клас, или многокласов дисбаланс. Например, при наличие на определено множество от данни с формат (X_i, y_i), където X_i е i^{th} наблюдение, то y_i е i^{th} класовият етикет, както следва $y_i \in \{1 \ldots K\}$ (Aly, 2005).

Koziarski, Wozniak и Krawczyk предлагат нова техника за свръхсемплиране наречен алгоритъм за комбинирано почистване и повторно вземане на проби (MC-CCR). Предложеният метод използва подход за моделиране на регионите подходящи за свръхизвадка, които са по-малко засегнати от раздеителни определения, *small disjuncts*, и отклонения при прилагане на техниката за свръхсемплиране на синтетичното малцинство, *SMOTE*. Целте е да се намали ефекта от припокриващите се класови разпределения върху производителността на алгоритмите за машинно самообучение (Koziarski, Wozniak, Krawczyk, 2020).

Експериментите им са базирани на 19 многокласови небалансирани множества, публично достъпни в базата с данни Knowledge Extraction based on Evolutionary Learning (KEEL, 2011). Koziarski, Wozniak и Krawczyk прилагат модела на дърветата на решенията (DT), модела базиран на теорията на вероятнстите и формулата на Бейс (Naïve-Bayes, NB както модела на на-близките съседи (K-Nearest Neighbor, KNN). Техният алгоритъм за комбинирано почистване и повторно вземане на проби, MC-CCR, е сравнен със съвременни методи за свръхсепмлиране на многокласови множества като техниката за свръхсемплиране на синтетичното малцинство (SMOTE-all, S-SMOTE), SMOTE комбиниран с итеративно-разделящ филтър (SMOTE-IPF), техниката за свръхсемплиране на разстояние Mahalanobis (Mahalanobis Distance Oversampling, MDO), както и техниката на свръхсемплиране на синтетично малцинство, *SMOM* (Koziarski, Wozniak, Krawczyk, 2020).

В настоящата секция разглеждаме мултикласовия вариант на алгоритъма за оптимизация на небалансирани множества, или алгоритъм за оптимизация на небалансирани множества с мулти класове (Data Centric Multiclass Optimization - DCMO). Резултатите постигнати от направените експеринети с DCMO ще бъдат съпоставени и сравнени с резлутатите представени от Koziarski, Wozniak и Krawczyk, постигнати от техния алгоритъм за комбинирано почистване и повторно вземане на проби (MC-CCR). За да осигурим правилно сравнение използваме същите множества и приложените от тях класифициращи алгоритми – DT, KNN, и NB.

Декларираме следните променливи:

- $i \in \{0 \ldots 100\}$
- r_i – random under-sampling integer
- $k_i \in \{1 \ldots 99\}$ – odd number generator
- $or_{i,k}$ – odd random dataset-split integer

- n – length of the given dataset m – minority class length
- R – list of integers OR – list of odd integers
- D_n – given dataset
- N_c – number of calsses in the dataset
- X_n – random variable (# of variables in D_n)
- Y_n – response variable (# of classes in D_n), Y = 1,…K, where K>=2
- $D_{i,m}$ – balanced, under-sampled data sub-set:
  $D_{i,m} = ([X_1,Y_1],….[X_n,Y_n] \mid r_i , m)$, where [X,Y] is independent of D_n
- $T_{i,k}$, $V_{i,k}$ – train and validation datasets
- score = 0 – optimal multicalss clasifier score metric
- min_optimal_score:
  - if desired minimal multicalss clasifier score:
    - min_optimal_score ∈ {0….100}
  - otherwise, None
- T_o and V_o – optimized train and validation sub-sets

ALGORITHM: *DCMO OPTIMIZATION PHASE*

Initialization of variables listed above.
set ***optimized*** *= False*
set ***odd_int_end*** *= False*
while *not* ***optimized***
  draw *random integer –* $\boldsymbol{r_i}$
  if $\boldsymbol{r_i}$ *not in list of integers (**R**)*
    append random integer (r_i*) to R*
    $\boldsymbol{D_{i,m}}$ *= undersample(*$D_n \mid r_i , m$*)*
    while *not* ***odd_int_end:***
      draw *odd random integer –* $\boldsymbol{or_{i,k}}$
      if $\boldsymbol{or_{i,k}}$ ***not*** *in list of off intergers (**OR**)*
        append odd random integer($or_{i,k}$*) to OR*
        split $D_{i,m}$ *into train and validation sets (80/20):*
        $\boldsymbol{T_{i,k}}$, $\boldsymbol{V_{i,k}}$ *= train_test_split (*$D_{i,m} \mid or_{i,k}$, *.20)*
        $\boldsymbol{C_{i,k}}$ *= build classifier*
        fit train set to $C_{i,k}$ *and calculate* F_1 *Score. Keep track:*
        $\boldsymbol{avg\ score_{i,k}}$ *=* $C_{i,k}\ (T_{i,k}, V_{i,k}) = \Sigma\ [(F1_{i,k} \mid C_{ik}, D_{i,m,k})]/Nc$
        evaluate current model $\boldsymbol{avg\ score_{i,k}}$
        if $\boldsymbol{avg\ score_{i,k}}$ *is greater than* ***score***
          score *= avg* $score_{i,k}$
          $\boldsymbol{T_o}$ *=* $T_{i,k}$
          $\boldsymbol{V_o}$ *=* $V_{i,k}$
      if lenght of OR is greater or equal than 50
      OR score >= optimal_score:
        odd_int_end *= True*
  if lenght of R is greater than 100
  OR score >= optimal_score:
    optimized *= True*
end

В Таблица 3.6.1, сравняваме резултатите, публикувани от Koziarski, Wozniak и Krawczyk с резултатите, получени от представения от нас алгоритъм за оптимизация на небалансирани множества с мулти класове (Data Centric Multiclass Optimization - DCMO).

Таблица 3.6.1

Множество	Results according to Average Accuracy (AvgAcc) [%] metric for MC-CCR and reference sampling methods with C5.0 as base classifier.									
							DCMO			
	MC-CCR	SMOTE-all	S-SMOTE	MDO	SMOM	SMOTE-IPF	AvgAcc	DT	KNN	NB
Automobile	76.98	80.12	73.53	78.13	79.04	75.32	**90.66**	96	88	88
Balance	82.87	55.06	55.01	57.70	59.52	54.26	**91.33**	90	87	97
Car	**97.12**	89.84	90.13	93.36	95.18	90.96	95.33	98	88	100
Cleveland	37.88	28.92	27.18	28.92	28.01	24.98	**87.33**	100	85	77
Contraceptive	53.18	50.63	46.92	53.27	55.09	52.88	**64.66**	70	65	59
Dermatology	94.29	95.72	96.1	97.48	99.31	92.18	**100**	100	100	100
Ecoli	**74.07**	64.68	67.54	61.16	61.16	60.43	N/A– not enough observations			
Flare	68.92	71.86	71.52	68.72	70.64	68.55	**79.66**	87	79	73
Hayes-Roth	92.11	86.45	88.04	87.33	90.06	89.74	**98.33**	100	95	100
Led7digit	70.48	72.39	72.55	75.03	75.94	71.35	**88.67**	91	91	84
Lymphography	**79.60**	73.02	62.67	76.54	74.72	74.20	N/A– not enough observations			
Newthyroid	96.18	94.7	93.48	92.06	90.24	93.05	**100**	100	100	100
Pageblocks	83.71	75.83	75.25	78.47	77.56	74.20	**95**	96	93	93
Thyroid	80.52	80.02	85.34	79.14	80.96	78.91	**90**	100	80	90
Vehicle	72.71	73.49	73.71	70.85	70.85	71.02	**78.67**	88	85	63
Wine	95.28	92.53	90.80	93.41	93.41	90.16	**97.00**	97	97	97
Winequality-red	46.93	37.41	35.79	40.05	42.78	36.28	**75.00**	75	75	92
Yeast	**58.39**	51.03	52.42	54.55	56.37	53.77	48.67	60	43	43
Zoo	**85.92**	82.61	68.69	79.09	79.09	67.30	N/A– not enough observations			

В Таблица 3.6.1, резултатите публикувани от Koziarski, Wozniak и Krawczyk са базирани на средна аритметична стойност на точност (AvgAcc) на MC-CCR метод. Резултатите от MC-CCR са сравнени с разгледаните от тях методи за свръхсепмлиране с основен класифицион алгоритъм DT (C5.0), като MC-CCR показва по добри стойности на точност от SMOTE, SMOTE-all, S-SMOTE, SMOTE-IPF, MDO и SMOM.

За да осигурим правилно сравнение между техния метод, MC-CCR, и представения от нас алгоритъм за оптимизация на небалансирани множества с мулти класове (Data Centric Multiclass Optimization - DCMO), ние предлагаме резултатите, постигнати от изброените алгоритми за машинно самообучение DT, KNN и NB както и средната аритметична стойност на точност (AvgAcc). Тук е важно да се отбележи съществената разлика между MC-CCR и DCMO, която се състои в третирането на разгледаните множества. MC-CCR прилага техниката за свръхсемплиране, *oversampling*, докато DCMO използва техниката на събсемплиране, *undersampling*. Това ограничава прилагането на DCMO върху три от разгледаните множества Ecoli, Lymphography и Zoo, тъй като един или повече от класовете имат прекалено нисък брой редове (наблюдения). В Таблица 3.6.1, тези множества са обозначени като не приложими, или *N/A– not enough observations*, както следва:

- Ecoli – клас imS,imL и omL имат 2, 2 и 5 наблюдения, съответно
- Lymphography – клас normal и fibrosis имат 2 и 4 наблюдения, съответно
- Zoo – клас 5,3 и 6 имат 4,5 и 8 наблюдения, съответно

Като разгледаме по-отблизо изброените мерки за останалите множества, можем да забележим, че предложеният от нас DCMO метод превъзхожда MC-CCR метода 73.7% (14/19) процента от времето базирано на AvgAcc. От друга страна, MC-CCR превъзхожда DCMO 26.3% (5/19) процента от времето. Както вече разяснихме по-горе, в три от случаите DCMO не може да бъде приложен поради много нисък брой на редове (наблюдения), т.е тук не може да се твърди със сигурност кой от двата метода би имал превес. Фактически, остават два случая в които MC-CCR превъзхожда DCMO 10.5% (2/19) процента от времето, в които AvgAcc е по-ниска. Това съвсем не е така, ако се вземе под внимание отделните резултати на приложените алгоритми за машинно обучение. Например, в Таблица 3.6.1 се виждат по-добри резлутати постигнати от DCMO за множествата Car и Yeast при прилагането на DT, както следва:

- Car:
  - DCMO – **98**
  - MC-CCR – 97.12
- Yeast:
  - DCMO – **60**
  - MC-CCR – 58.29

3.2.1 Тестови мерки за цялостно оценяване на алгоритъма за оптимизация на небалансирани множества с мулти класове (Data Centric Multiclass Optimization - DCMO)

В допълнение на тестови мерки за цялостно оценяване на модели, а именно прецизност (Precision), рикол (Recall), f1 показател (F1 Score, F1), коефициент на корелация на Matthew (MCC), както и матрицата с допуснати грешки от моделите, *confusion matrix*, в следващите секции ние отчитаме тестовите резултати получени от DCMO при прилагане на алгоритмите за DT, KNN и NB за класификационен анализ.

Разглеждайки Таблица 3.2.1, се наблюдават два случая в които DCMO дава ниски резутати при прилагането на *Yeast* и *Contraceptive* множествата. *Yeast* множеството съдържа 9 класа, от които класът на малцинството има едва 20 наблюдения. Това означава, че при прилагане на стандартното съотношение за разделение на тренировъчно и тестово подмножества от 80/20, DCMO ще оптимизира и балансира *Yeast* множеството до 4 наблюдения за всеки от 9 класа в тестовото подмножество (Приложение – 3.2.1.16). DCMO постига незадоволяващия резултат от едва 60 процента точност (MC-CCR точност - 58.39). Тук се здава въпроса достатъчнии ли са едва 16 наблюдения за всеки клас, за да се достигне до оптимална точност при машинното самообучение?

Таблица 3.2.1 Резултати получени от DCMO за разгледаните множества.

Множества	Клас на малцинството (minority class)	Алгоритъм	DCMO средно аритметични тестови мерки (Macro average)				
			Accuracy	Precision	Recall	F1	MCC
Automobile	Клас 4	DT	96	97	96	96	95
		KNN	88	91	88	88	86
		NB	88	91	88	88	86
Balance	Клас 1	DT	90	91	90	90	85
		KNN	87	87	87	87	89
		NB	97	97	97	97	95
Car	Клас 3	DT	98	98	98	98	97
		KNN	88	89	88	88	85
		NB	100	100	100	100	100
Cleveland	Клас 4	DT	100	100	100	100	100
		KNN	85	88	83	85	82
		NB	77	72	73	77	72
Contraceptive	Клас 1	DT	70	69	70	70	55
		KNN	65	64	64	64	47
		NB	59	62	59	59	41
Dermatology	Клас 5	DT	100	100	100	100	100
		KNN	100	100	100	100	100
		NB	100	100	100	100	100
Flare	Клас 4	DT	87	86	86	85	84
		KNN	79	81	79	79	75
		NB	73	78	73	71	69
Hayes-Roth	Клас 2	DT	100	100	100	100	100
		KNN	95	96	94	95	92
		NB	100	100	100	100	100
Led7digit	Клас 1	DT	91	92	91	91	90
		KNN	91	92	91	91	90
		NB	84	85	84	83	82
Newthyroid	Клас 2	DT	100	100	100	100	100
		KNN	100	100	100	100	100
		NB	100	100	100	100	100
Pageblocks	Клас 2	DT	96	97	97	96	96
		KNN	93	94	93	93	91
		NB	93	95	93	93	92
Thyroid	Клас 0	DT	100	100	100	100	100
		KNN	80	81	80	80	70
		NB	90	91	90	90	86
Vehicle	Клас 0	DT	88	89	88	88	84
		KNN	85	85	85	85	80
		NB	63	68	63	62	53

Wine	Клас 2	DT	97	97	97	97	95
		KNN	97	97	97	96	95
		NB	97	97	96	96	95
Winequality-red	Клас 0	DT	75	64	75	75	72
		KNN	75	75	75	72	72
		NB	92	94	92	91	91
Yeast	Клас 0	DT	60	62	60	59	55
		KNN	43	32	44	36	37
		NB	43	52	43	41	38

По скоро, отговорът се крие в качеството на разглежданите множества от данни. Например, както се вижда в горната Таблица 3.2.1, *Contraceptive* множеството има 3 класа, от които класът на малцинството има 333 наблюдения, следователно, тестовото *Contraceptive* подножество се състои от 67 (20% от 333) наблюдения за всеки клас (Приложение - 3.2.1.5). DCMO постига незадоволяващия резултат от 70 процента точност, който е едва с 10 процента по-добър от експеримента с Yeast множеството, независимо от голямата разлика в тренировъчните наблюдеиня от ~17 пъти.

От друга страна, множеството *Dermatology*, подобно на *Yeast*, съдържа 6 класа, от които класът на малцинството има 20 наблюдения, следователно, тестовото *Dermatology* подножество се състои от 4 наблюдения за всеки клас (Приложение - 3.2.1.6). DCMO постига точност от 100 процента и за трите разглеждани алгоритми за машинно самообучение - модела на дърветата на решенията (DT), модела, базиран на теорията на вероятностите и формулата на Бейс (Naïve-Bayes, NB както и модела на на-близките съседи (K-Nearest Neighbor, KNN).

Подобни резултати се наблюдават и при *Newthyroid* множеството. *Newthyroid* съдържа 3 класа, от които класът на малцинството има 30 наблюдения, следователно, тестовото *Newthyroid* подножество се състои от 6 (20% от 30) наблюдения за всеки клас (Приложение - 3.2.1.10). DCMO постига точност от 100 процента и за трите разглеждани алгоритми за машинно самообучение - DT, NB и KNN. Тези експерименти показват, че големият размер на множествата от данни не винаги водят до задоволяваща точност при машинното обучение. Оптимизирането и балансирането на множествата, *Data-centric optimization,* води до подобряване на ефективността на проблемите с класовият дисбаланс, състоящи се от клас на мнозинството (по-голям брой наблюдения) и клас на малцинството.

Основни приноси

В настоящата дисертация са изследвани проблеми в две основни направления:

- Извеждане и изследване на итерационни методи за търсене на равновесие на Неш в динамични игри. Изследванията са описани в глава първа и втора;
- Създаване на модели и алгоритми за провеждане на класификационен анализ на конкретни множества. Изследванията са описани в трета глава.

Публикации по дисертационния труд на Владислав Танов:

1. Ivan Ivanov, Nikolay Netov, **Vladislav Tanov**, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games. *International Journal of Mathematical and Computational Methods*, **1**, 378-381, 2016.
2. Ivelin G. Ivanov, **Vladislav Tanov**, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, *Mathematics, and its Applications / Annals of AOSR*, 10(2), 202-210, 2018. (Scopus)
3. Ivelin G. Ivanov, **Vladislav Tanov**, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, *Mathematics and its Applications / Annals of AOSR*, 10(2), 230-244, 2018. (Scopus)
4. Ivelin G. Ivanov, **Vladislav Tanov**, A Nonsymmetric Nash-Riccati Equation and Decoupled Schemes for a Stabilizing Solution, *Applied Mathematics E - Notes*, 2020, 20, pp. 357–366. (Scopus)
5. **Vladislav Tanov**, Ivan Ivanov, Data Centric Optimization Method to Imbalanced Datasets, *Proceedings of SPIE - The International Society for Optical Engineering*, 12616, 1261602, 2023. (Scopus), International Conference on Mathematical and Statistical Physics, Computational Science, Education, and Communication (ICMSCE 2022), 2022, Istanbul, Turkey. (Scopus) https://doi.org/10.1117/12.2674455 (Scopus)
6. **Vladislav Tanov**, Data-Centric Optimization Approach for Small, Imbalanced Datasets, *Journal of Information and Organizational Sciences (JIOS)*, Vol 47, No 1, 2023. https://jios.foi.hr/index.php/jios/article/view/1875 (Scopus)

БЛАГОДАРНОСТ

Благодаря на научния си ръководител доц.д-р Николай Нетов за споделените знания по темите на дисертацинния труд и неоценимата помощ в хода на обучението и написването на дисертационния труд.

Литература

1 Иван Иванов, Обобщени Рикатиеви уравнения в стохастичното икономическо моделиране, УИ „Св. Кл. Охриски“, София, 2012.

2 Ивелин Иванов, ИНФОРМАЦИОННИ ТЕХНОЛОГИИ ЗА УПРАВЛЕНИЕ НА ПРОЦЕСИ (МОДЕЛИ), дисертационен труд, Шуменски университет 2016.

3 M**.** Aly**,** Survey on multiclass classification methods. *Technical Report, Caltech*, 2005

4 K. Azbeg, M. Boudhane, O. Ouchetto, Diabetes emergency cases identification based on a statistical predictive model, *Journal of Big Data*, 2022

5 T.Azevedo-Perdicoulis, G.Jank, Linear Quadratic Nash Games on Positive Linear Systems, European Journal of Control, 11, 1-13, 2005.

6 B. Basar, G.J. Olsder. Dynamic Noncooperative Game Theory. SIAM, Philadelphia, 1999.

7 Z.-Z. Bai, X.-X. Guo, S.-F. Xu, alternately linearized implicit iteration methods for the minimal nonnegative solutions of the nonsymmetric algebraic Riccati equations, Numer. Linear Algebra Appl., 13, 655-674, 2006.

8 N.Baeva, The Nash Equilibrium in Open Loop Linear Quadratic Games for Positive Systems, Mathematics and its Applications / Annals of AOSR, 9(1), 17-27, 2017.

9 T. Bayes. An essay towards solving a Problem in the Doctrine of Chances. Bayes's essay as published in the Philosophical Transactions of the Royal Society of London, Vol. 53, p. 370. 1763.

10 J. Bohacik and M. Zabovsky, Discretization for Naive Bayes Taking the Specifics of Heart Data into Account. Journal of International and Organizational Science, vol. 43, no. 1, 2019

11 D. Berrar. Bayes’ Theorem and Naive Bayes Classifier. *Research Gate*. DOI: 10.1016/B978-0-12-809633-8.20473-1. 2018.

12 P. Branco, L. Torgo, R.P. Ribeiro. Relevance-based evaluation metrics for multi-class imbalanced domains. *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference*. Proceedings, Part I, 2017, pp. 698–710. 2017.

13 L. Breiman, J.H. Friedman, C.J. Stone, R.A. Olshen. Classification and Regression Trees. *Wadsworth & Brooks*. 1984.

14 Breiman, L. Bagging predictors. Mach Learn 24. 1996.

15 Breiman L. Random Forests. Machine Learning. 2001.

16 L. M. Bruce, Game theory applied to big data analytics in geosciences and remote sensing, 2013 IEEE International Geoscience and Remote Sensing Symposium, https://ieeexplore.ieee.org/document/6723733 (2018)

17 T. Carter. An introduction to information theory and entropy. *Complex Systems*. 2011

18 R.T. Cox. Probability, Frequency, and Reasonable Expectation. American Journal of Physics. 1946.

19 L.S. Chen, S.J. Cai, Neural-network-based resampling method for detecting diabetes mellitus, SpringerLink, 2015
20 D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation., *BMC Genomics*, 2020
21 O.L.V. Costa , W.L. de Paulo, Indefinite quadratic with linear costs optimal control of Markov jump with multiplicative noise systems, Automatica, Vol. 43, 587-597, 2007. (Costa , de Paulo, 2007)
22 O.L.V. Costa, A. de Oliveira, Optimal mean-variance control for discrete-time linear systems with Markovian jumps and multiplicative noises, Automatica, 48, 2012, 304-315. (Costa, de Oliveira, 2012).
23 CDC, Heart Disease | Cdc.Gov, *Centers for Disease Control and Prevention*, www.cdc.gov, 2022.
24 T. Damm, D. Hinrichsen, , Newton's Method for a Rational Matrix Equation occurring in stochastic control, Linear Algebra Appl., 332-334:81-109., 2001 (Damm and Hinrichsen 2001)
25 E. Dockner, S. Jorgensen, N. V. Long, G. Sorger, Differential games in economics and management science, Cambridge University Press, (2000).
26 A. K. Dubey, A. K. Sinhal, R. Sharma, An Improved Auto Categorical PSO with ML for Heart Disease Prediction, *Engineering, Technology and Applied Science Research*, 2022
27 V. Dragan, S. Aberkane, I. Ivanov, An iterative procedure for computing the stabilizing solution of discrete-time periodic Riccati equations with an indefinite sign, 21st International Symposium on Mathematical Theory of Networks and Systems July 7-11, 2014. Groningen.
28 V. Dragan, S. Aberkane, I. Ivanov, On computing the stabilizing solution of a class of discrete-time periodic Riccati equations, International Journal of Robust and Nonlinear Control, vol.25, 7, 2015, 1066-1093, doi: 10.1002/rnc.3131.
29 EMC. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. *Wiley*. 2015. https://www.wiley.com/en-ie/Data+Science+and+Big+Data+Analytics%3A+Discovering%2C+Analyzing%2C+Visualizing+and+Presenting+Data-p-9781118876138
30 H. J. Escalante, M. Montes, L. E. Sucar, Practicle Swarm Model Selection, *Journal of Machine Learning*, 2009
31 J. Engwerda. LQ Dynamic Optimization and Differential Games, Wiley 2005.
32 J. Grus. Data Science from Scratch. *O'Reilly Media*. ISBN: 9781491901427. 2015.
33 U. M. Fayyad and K. B. Irani, Multi-Interval discretization of continuous-valued attributes for classification learning, *International Joint Conference on Uncertainty in AI*, 1993.
34 F. Farris. The Gini Index and Measures of Inequality. *American Mathematical Monthly*. 117. 2010.

35 Y.T. Feng, B.D.O. Anderson, An iterative algorithm to solve state-perturbed stochastic algebraic Riccati equations in LQ zero-sum games, Systems & Control Letters, 59(1)(2010), 50–56.
36 B. de Finetti. Theory of Probability: A critical introductory treatment. Chichester: John Wiley & Sons Ltd. 2017.
37 G. Freiling, A.Hochhaus, On a Class of Rational Matrix Differential equations Arising in Stochastic Control. Linear Algebra Its App., 379, 43-68, 2004.
38 FastStats. *FastStats - Chronic Liver Disease or Cirrhosis*, www.cdc.gov, 2022.
39 L. Imsland, I. G. Ivanov, and S. Kostova, Linear Quadratic Differential Games and Applications, Biomath Communications, 3, 2016, N 2.
40 I. G. Ivanov, L. Imsland and B. C. Bogdanova, Iterative algorithms for computing the feedback Nash equilibrium point for positive systems, International Journal of System Science.
41 I. G.Ivanov and N. Netov, The Nash Equilibrium Point in the LQ Game on Positive Systems with Two Players, Mathematical and Computational Methods, 1, 2016, 242–246.
42 Ivan Ivanov, Nikolay Netov, Vladislav **Tanov**, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games. *International Journal of Mathematical and Computational Methods*, **1**, 378-381, 2016
43 Ivelin Ivanov, Vladislav **Tanov**, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, Mathematics and its Applications / Annals of AOSR, 10(2), 230-244, 2018.
44 Ivelin Ivanov, Vladislav **Tanov**, A Nonsymmetric Nash-Riccati Equation and Decoupled Schemes for a Stabilizing Solution, Applied Mathematics E-Notes, 20(2020), 357-366, Applied Mathematics E-Notes, 20(2020), 357-366
45 Ivelin G. Ivanov, Vladislav **Tanov**, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, Mathematics and its Applications/Annals of AOSR, 10(2), 202-210, 2018.
46 G. Jank, D. Kremer, Open loop Nash games and positive systems solvability conditions for non symmetric Riccati equations.
47 E.T. Jaynes. Bayesian Methods: General Background. In Justice, J. H. (ed.). Maximum-Entropy and Bayesian Methods in Applied Statistics. Cambridge: Cambridge University Press. 1986.
48 M.A. Hall, Correlation-based feature selection for machine learning, 1999.
49 H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *International Joint Con- ference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE. 2008
50 S. Jorgensen, and G. Zaccour, Differential Games in Marketing. International Series in Quantitative Marketing, Kluwer Academic Publisher in 2004.
51 M. Koziarski, M. Wozniak. CCR: A combined cleaning and resampling algorithm for imbalanced data classification. *Appl. Math. Comput. Sci*. 2017.

52 M. Koziarski**,** M. Wozniak, B. Krawczyk, Combined Cleaning and Resampling algorithm for multi-class imbalanced data with label noise, *Knowledge-Based Systems*, 2020
53 B. Krawczyk, M. Koziarski, M. Wozniak. Radial-Based Oversampling for Multiclass Imbalanced Data Classification, *IEEE Trans. Neural Netw. Learn. Syst*. 2019.
54 A. Lanzon, Feng Y., Anderson B., Rotkowitz M., Computing the positive stabilizing solution to algebraic Riccati equations with an indefinite quadratic term via a recursive method, IEEE Transactions on automatic control, 53,10(2008), 2280-2291.
55 T.H. Lee, A. Ullah, R. Wang. Bootstrap Aggregating and Random Forest. UC Riverside Department of Economics. 2019.
56 D.J.N.Limebeer, B.D.O.Anderson, B.Hendel. A Nash game approach to mixed H_2/H_{\infty} control. IEEE Transactions on Automatic Control, 1994, 39(1), 69-82, 1994.
57 C.Ma, H.Lu. Numerical Study on Nonsymmetric Algebraic Riccati Equations, Mediterranean J. of Mathematics, 13, 6, 4961-4973, 2016.
58 A. McCallum. Graphical Models, Lecture2: Bayesian Network Representation. *Archived (PDF)*. 2022.
59 T.Mitchell. Machine Learning. *McGraw-Hill Science*. ISBN: 0070428077. 1997
60 L. Metzler, Stability of multiple markets: the Hicks condition, Econometrica, 13(4), 1945, pp. 277–292
61 Z. Mo, D. Xuan, R. Shi, 2023, Robust Data Sampling in Machine Learning: A Game-Theoretic Framework for Training and Validation Data Selection, *Games* 14, no. 1: 13. https://doi.org/10.3390/g14010013
62 N. Moniz, P. Branco, L. Torgo, Resampling strategies for imbalanced time series forecasting, SpringerLink, 2017.
63 D.M.B. Powers**,** Evaluation: From Precision, Recall and F-Score to ROC, Informedness, Markedness & Correlation, *Journal of Machine Learning Technologies*, 2011.
64 Q. Qiao, A.Y. Kaltungo, R.e. Edwards, Feature selection strategy for machine learning methods in building energy consumption prediction, *Science Direct*, vol. 8, 2022.
65 J. R. Quinlan. Induction of Decision Trees. Mach. Learn. 1. 81–106. 1986.
66 J. R. Quinlan. C4.5: Programs for Machine Learning. *Morgan Kaufmann Publishers*, 1993
67 J. R. Quinlan. Improved use of continuous attributes in 4.5. Journal of Artificial Intelligence Research. 4:77-90. 1996.
68 B. Raskutti, A. Kowalczyk. Extreme rebalancing for svms: a case study, SIGKDD Explorations, 2004
69 S. Russell, P. Norvi. Artificial Intelligence: A Modern Approach. *Prentice Hall*. p. 478. 2002.
70 W. Satriaji, R. Kusumaningrum, Effect of Synthetic Minority Oversampling Technique (SMOTE), Feature Representation, and Classification Algorithm on

Imbalanced Sentiment Analysis, 2nd International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2018.

71 A. Samuel, Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development *3* (3). 1959.

72 D. Schum. The Evidential Foundations of Probabilistic Reasoning. Northwestern University Press. 1994.

73 B. Sjardin, L. Massaron, A. Boschetti. Large Scale Machine Learning with Python. *Packt Publishing*. 2016.

74 N. Shaltout, M. Elhefnawi, A. Rafea and A. Moustafa, Information Gain as a Feature Selection Method for the Efficient Classification of Influenza Based on Viral Hosts, Engineering and Computer Science, 2014.

75 C.E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*. 27. 379–423. 1948.

76 J. Singh, S. Bagga and R. Kaur, Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques, Procedia Com. Science, 2020

77 UCI Machine Learning Repository: ILPD (Indian Liver Patient Dataset) Data Set. *UCI Machine Learning Repository: ILPD (Indian Liver Patient Dataset) Data Set*, archive.ics.uci.edu

78 UCI Machine Learning Repository: Statlog (Heart) Data Set. UCI Machine Learning Repository: Statlog (Heart) Data Set.

79 H.Zhu, C.Zhang, Infinite time horizon nonzero-sum linear quadratic stochastic differential games with state and control-dependent noise, J. Control Theory Appl., 11: 629-633, 2013.

80 H.N. Zhu, C.K. Zhang, N. Bin, Stochastic Nash Games for Markov Jump Linear Systems with State- and Control-Dependent Noise, Journal of the Operations Research Society of China, 2: 481-498, 2014.

81 D. Yao, S. Zhang X. Zhou, Stochastic linear quadratic control via semidefinite programming, SIAM Journal Control Optimization, 49 (3):801-823, 2001.

82 J. VanderPlas. Python Data Science Handbook: Essential Tools for Working with Data 1st Edition, O'Reilly Media; 1st edition. ISBN: 1491912057, 2017.

83 J. Wainer, G. Cawley, Empirical evaluation of resampling procedures for optimising SVM hyperparameters, J. of Machine Learning Research, 18, 2017.

84 K. Wang, W. Sun, Q.Du, A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation. Com. Mech 64, 467–499(2019). https://doi.org/10.1007/s00466-019-01723-1

85 G.M. Weiss**,** H. Hirsh, A Quantitative Study of Small Disjuncts. *In Proceedings of the Seventeenth National Conference on Artificial*. 2000

86 H. Witten, E. Frank, M.A. Hall. Data Mining: Practical machine learning tools and techniques, 3rd Edition. *Morgan Kaufmann*. p. 191. 2011.

87 H. Written**,** E. Frank, A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, *Morgan Kaufmann*, 4th ed. Burlington, MA, 2016.

88 G. Wu, E. Chang. Lecture, Topic. Class-Boundary Alignment for Imbalanced Dataset Learning, ICML Workshop on Learning from Imbalanced Data Sets II, Washington, DC, 2003.